UC Santa Barbara

UC Santa Barbara Electronic Theses and Dissertations

Title

Towards Segment-level Video Understanding: Detecting Activities from Untrimmed Videos

Permalink

https://escholarship.org/uc/item/5vn7r9kd

Author

Zhang, Da

Publication Date

2020

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

University of California
Santa Barbara

Towards Segment-level Video Understanding: Detecting Activities from Untrimmed Videos

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Computer Science

by

Da Zhang

Committee in charge:

Professor Yuan-Fang Wang, Chair
Professor Matthew Turk
Professor Xifeng Yan

March 2020

The Dissertation of Da Zhang is approved.

Professor Matthew Turk

Professor Xifeng Yan

Professor Yuan-Fang Wang, Committee Chair

December 2019

Towards Segment-level Video Understanding:

Detecting Activities from Untrimmed Videos

Copyright © 2020

by

Da Zhang

This dissertation is dedicated to

my fiancee Jing Li

for her endless love, support and encouragement;

my family, friends and loved ones

who helped me in all things great and small.

Acknowledgements

There are many people that have earned my gratitude for their contribution to my time in graduate school without whom this thesis would not have been possible.

First, I would like to thank my thesis advisor Prof. Yuan-Fang Wang. Yuan-Fang believed in me like nobody else and gave me endless support. On the academic level, Yuan-Fang taught me fundamentals of conducting scientific research in the areas of computer vision and deep learning. On the personal level, Yuan-Fang inspired me by his hardworking and passionate attitude. For all these, I would give my sincere gratitude to Prof. Yuan-Fang Wang.

Second, I would also like to thank my committee members, Prof. Matthew Turk and Prof. Xifeng Yan, for providing valuable feedback and advise during my PhD. I am thankful to Prof. Matthew Turk, an expert in computer vision, for his crucial remarks that shaped my final dissertation. I am also grateful to Prof. Xifeng Yan for his insightful comments and for sharing with me his tremendous academic experiences.

Third, it was my greatest pleasure working with my great collaborators. This dissertation would not have been possible without the intellectual contribution of Xiyang Dai, a PhD researcher from University of Maryland. Moreover, I am thankful to my lab mate Xin Wang for his collaboration and contribution in various projects related to this dissertation. I would also like to thank my other lab mates for making my experience in the computer vision lab and graduate school exciting and fun.

I am also grateful to my industrial collaborators and mentors. I have spent three great summers at Samsung Research America, Stanford Research International and Amazon where I had the chance to collaborate with fantastic researchers. I would like to thank Hamid Maei, Yi Yao and Manoj Aggarwal for their great mentorship and guidance. I also extend my gratitude to the group members for the fruitful discussions and for making

my internship such an eye-opening experience.

Last but not least, I would like to express my deepest gratitude to my family and friends. This dissertation would not have been possible without their warm love, continuous patience, and unconditional support.

Curriculum Vitæ

Da Zhang

Education

2020	Ph.D. in Computer Science, University of California, Santa Barbara.
2014	M.S. in Electrical Engineering, Shanghai Jiao Tong University, China.

Experience

10/2014 - 03/2020	Ph.D. Researcher, University of California, Santa Barbara
06/2019 - 09/2019	Applied Scientist Intern, Amazon Go
06/2017 - 09/2017	Research Intern, Stanford Research International
06/2016 - 09/2016	Research Intern, Samsung Research America

Publications

"METAL: Minimum Effort Temporal Activity Localization in Untrimmed Videos". **Da Zhang**, Xiyang Dai, Yuan-Fang Wang, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, USA, June 2020.

"MAN: Moment Alignment Network for Natural Language Moment Retrieval via Iterative Graph Adjustment". **Da Zhang**, Xiyang Dai, Xin Wang, Yuan-Fang Wang, Larry Davis, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, USA, June 2019.

"Dynamic Temporal Pyramid Network: A Closer Look at Multi-Scale Modeling for Activity Detection". **Da Zhang**, Xiyang Dai, Yuan-Fang Wang, Proceedings of the Asian Conference on Computer Vision (ACCV Oral), Perth, Australia, December 2018.

"S3D: Single Shot multi-Span Detector via Fully 3D Convolutional Network". **Da Zhang**, Xiyang Dai, Xin Wang, Yuan-Fang Wang, Larry Davis, Proceedings of the British Machine Vision Conference (BMVC Oral), Newcastle upon Tyne, UK, September 2018.

"Deep Reinforcement Learning for Visual Object Tracking in Videos". **Da Zhang**, Hamid Maei, Xin Wang, Yuan-Fang Wang, ArXiv preprint. 2017.

"Learning to Compose Topic-Aware Mixture of Experts for Zero-Shot Video Captioning". Xin Wang, Jiawei Wu, **Da Zhang**, Yu Su, William Yang Wang, Proceedings of the AAAI Conference on Artificial Intelligence (AAAI Oral), Honolulu, USA, January 2019.

"Multimodal Transfer: A Hierarchical Deep Convolutional Neural Network for Fast Artistic Style Transfer". Xin Wang, Geoffrey Oxholm, **Da Zhang**, Yuan-Fang Wang, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, USA, June 2017.

"INDAPSON: An Incentive Data Plan Sharing System Based on Self-Organizing Network". Tuo Yu, Zilong Zhou, **Da Zhang**, Xinbing Wang, Yunxin Liu, Songwu Lu, Proceedings of the IEEE Conference on Computer Communications (INFOCOM), Toronto, Canada, April 2014.

Abstract

Towards Segment-level Video Understanding:
Detecting Activities from Untrimmed Videos

by

Da Zhang

We generate massive amounts of video data every day. While most real-world videos are long and untrimmed with sparsely localized segments of interest, existing AI systems that can interpret videos today often rely on static image analysis or can only process temporal information in a short video snippet. To automatically understand the content of long video streams, this thesis mainly describes the efforts to design accurate, efficient, and intelligent deep learning algorithms for temporal activity detection in untrimmed videos.

Detecting segments of interest from untrimmed videos is a key step towards segment-level video understanding. Depending on the purposes of tasks being performed, we address three different activity detection tasks: detecting activities of interest from videos without specific purposes (*i.e.*, temporal activity detection); detecting temporal segment that best corresponds to a language query (*i.e.*, natural language moment retrieval); and detecting activities given less supervision (*i.e.*, weakly-supervised or few-shot activity detection).

In temporal activity detection, We first propose a highly unified single-shot temporal activity detector based on fully 3D convolutional networks, by eliminating explicit temporal proposal and classification stages. Evaluations show that it achieves state-of-the-art on temporal activity detection while being super efficient to operate at 1271 FPS. We then investigate how to effectively apply a multi-scale architecture to model activities

with various temporal length and frequency. We propose three novel architecture designs: (1) dynamic temporal sampling; (2) two-branch feature hierarchy; (3) multi-scale contextual feature fusion, and we combine all these components into a uniform network and achieve the state-of-the-art on a much larger temporal activity detection benchmark.

In natural language moment retrieval, we aim to localize the segment that best corresponds to a given language query. We present a language-guided temporal attention module and an iterative graph adjustment network to handle the semantic and structural misalignment between video and language. The proposed model demonstrates superior capability to handle temporal relations, thus, significantly improves the state-of-the-art by a large margin.

Finally, we study the problem of weakly-supervised and few-shot temporal activity detection to mitigate the drawbacks of huge amounts of supervision needed to train a temporal detection model. Namely, we answer the question if we can learn a temporal activity detector under weak supervision that is able to localize unseen activity classes. A novel meta-learning based detection method is accordingly proposed by adopting the few-shot learning technique of Relation Network. Results show that our method achieves performance superior or competitive to state-of-the-art approaches with stronger supervision.

In summary, we propose a suite of algorithms and solutions to automatically detect segments of interest in long untrimmed videos. We hope our studies could provide insights for researchers to explore new deep learning paradigms for future computer vision research, especially on video-related topics.

Contents

Curriculum Vitae **vii**

Abstract **ix**

List of Figures **xiv**

List of Tables **xviii**

1 Introduction **1**

2 Related Work **5**
2.1 Activity Recognition 5
2.2 Object Detection and Multi-Scale Modeling 6
2.3 Temporal Activity Detection 7
2.4 Natural Language Moment Retrieval 8
2.5 Visual Relations and Graph Network 9
2.6 Weakly Supervised Detection 10
2.7 Few-Shot Learning 10

3 Single Shot Multi-Span Detector via Fully 3D Convolutional Network **12**
3.1 Introduction 13
3.2 Single-Shot multi-Span Detector 15
3.2.1 Base Feature Layers 15
3.2.2 Auxiliary Temporal Feature Layers 16
3.2.3 Multi-scale Default Spans 16
3.2.4 Convolutional Predictors 17
3.3 Network Training 18
3.3.1 Training Data Construction and Augmentation 19
3.3.2 Matching Strategy 19
3.3.3 Hard Negative Mining 20
3.3.4 Training Objective 20
3.3.5 Prediction 22

3.4 Experiments . . . 22
3.4.1 Comparison with State-of-the-art . . . 23
3.4.2 Ablation Study . . . 24
3.4.3 Qualitative Results . . . 26
3.5 Conclusion . . . 27

4 Dynamic Temporal Pyramid Network for Temporal Multi-Scale Modeling 28
4.1 Introduction . . . 29
4.2 Dynamic Temporal Pyramid Network . . . 31
4.2.1 Pyramidal Input Feature Extraction with Dynamic Sampling . . . 32
4.2.2 Multi-scale Feature Hierarchy with Two-branch Network . . . 34
4.2.3 Local and Global Temporal Contexts . . . 37
4.3 Experiments . . . 38
4.3.1 Implementation Details . . . 39
4.3.2 Comparison with State-of-the-art . . . 41
4.3.3 Ablation Study . . . 42
4.3.4 Qualitative Results . . . 44
4.4 Conclusion . . . 46

5 Moment Alignment Network for Natural Language Moment Retrieval 47
5.1 Introduction . . . 48
5.2 Moment Alignment Network . . . 50
5.2.1 Language Encoding as Dynamic Filters . . . 51
5.2.2 Single-Shot Video Encoder . . . 53
5.2.3 Iterative Graph Adjustment Network . . . 55
5.2.4 Network Training . . . 57
5.3 Experiments . . . 58
5.3.1 Comparison with State-of-the-art . . . 60
5.3.2 Ablation Studies . . . 61
5.3.3 Visualization . . . 66
5.4 Conclusion . . . 66

6 Similarity Pyramid Network for Minimum Effort Temporal Activity Localization 68
6.1 Introduction . . . 69
6.2 Similarity Pyramid Network . . . 72
6.2.1 Model Overview . . . 73
6.2.2 Video Embedding Module . . . 74
6.2.3 Temporal Feature Pyramid . . . 75
6.2.4 Multi-scale Relation Module . . . 76
6.2.5 Training . . . 77
6.2.6 Prediction . . . 80

6.3 Experiments 80
6.3.1 Dataset and Evaluation 80
6.3.2 Comparison with State-of-the-art 83
6.3.3 Ablation Studies 85
6.3.4 Visualization 86
6.4 Conclusion 90

7 Conclusion **92**
7.1 Summary of Contributions 92
7.2 Future works 95
7.2.1 Large-scale segment retrieval 95
7.2.2 Spatial temporal video understanding 95

Bibliography **97**

List of Figures

3.1 S^3D network architecture: Our network takes a video of 256 frames with spatial size 112×112 as input and computes base features using a standard C3D [1] network up to conv5b. We add auxiliary Conv3D layers on top of conv5 to produce a temporal feature hierarchy with multi-scale default spans at each layer. For each temporal feature map cell, we predict K class confidence scores, 1 activity confidence score and 2 location offsets with a set of Conv3D filters. Temporal NMS is applied to produce the final detection results. 14

3.2 S^3D framework. (a) Input video with temporal ground-truth annotations. We evaluate a small set (*e.g.* 4) of multi-scale default spans at each location in several feature maps with different temporal resolutions (*e.g.* conv7 in (b) and conv8 in (c)). For each default span, we predict both the temporal offsets and the confidences for presence of activity and all activity categories. At training time, we match the default spans to the ground truth spans. 18

3.3 Qualitative visualization of the top detected activities by S^3D (best viewed in color) on four different activity categories in THUMOS'14 dataset: *Pole Vault*, *Clean and Jerk*, *Javelin Throw* and *Shotput*. Ground truth activity segments are marked in black and predicted activity segments are marked in green. 26

4.1 An illustration of pyramidal input feature extraction with 5 sampling rates. Left: input video is sampled at different FPS to capture motion dynamics at different temporal resolutions; Right: a shared 3D ConvNet is used to extract the input feature at each resolution. 32

4.2 An illustration of the two-branch multi-scale network with $S = N = 5$. The network combines a temporal convolution branch and a temporal pooling branch, where the features are concatenated and down sampled. Late fusion scheme is applied to build the multi-scale feature hierarchy. . 35

4.3 An illustration of local and global contexts when setting N to 5. Every temporal feature map cell at all scales is enhanced by its local context (next scale) and global context (last scale) to produce a set of prediction parameters. Temporal NMS is applied to produce the final detection results. 37
4.4 Qualitative visualization of the top detected activities on ActivityNet. Each sequence consists of the ground-truth (blue) and predicted (green) activity segments and class labels. 45

5.1 The natural language moment retrieval task in untrimmed videos. To properly localize the moment, the retrieval model must handle both *semantic misalignment (top)* with multiple moments of interests and *structural misalignment (bottom)* with complex temporal dependencies. 48
5.2 An overview of our end-to-end Moment Alignment Network (MAN) for natural language moment retrieval (best viewed in color). MAN consists three major components: (1) A language encoder to convert the input language query to dynamic convolutional filters through a single-layer LSTM. (2) A video encoder to produce multi-scale candidate moment representations in a hierarchical fully-convolutional network, where input visual features are aligned with language semantics by convolution. (3) An iterative graph adjustment network to directly model moment-wise temporal relations and update moment representations. Finally, the moments are retrieved by its matching scores with the language query. 51
5.3 The structure of the proposed Iterative Graph Adjustment Network (IGAN). Top: In each IGAN cell, a residual component R_t is generated from the previous node representation X_{t-1} and aggregated with the preserving component G_{t-1} to produce the current adjacency matrix G_t. Node representations are updated according to Equation 5.3 with G_t, X_0 and W_t^o. Bottom: Multiple IGAN cells are connected to simultaneously update node representation and graph structure. 57
5.4 Qualitative visualization of the natural language moment retrieval results (Rank@1) by MAN (best viewed in color) on four different video-query pairs in Charades-STA dataset. Ground truth moments are marked in black and retrieved moments are marked in green. MAN is able to retrieve single moments such as "takes a bite" and "smiling" and continuous moments such as "gets into a desk" followed by "runs into a room" and "takes off the shoes" followed by "walks by". 64

5.5 Qualitative visualization of the natural language moment retrieval results (Rank@1) by MAN (best viewed in color) on four different video-query pairs in DiDeMo dataset. Ground truth moments are marked in black and retrieved moments are marked in green. MAN is able to retrieve moments described by complex temporal dependencies such as "for the first time" and "after", and it can also distinguish the desired moments from similar unrelated contexts like correctly identify "cross the view" moment and the moment when "you can see his belly button". 65
5.6 Qualitative example of MAN evaluated on a video-query pair (best viewed in color). The final moment-wise graph structure with top related edges and their corresponding moments is visualized. The retrieved moment is marked in green and other moments are marked in blue. The dashed line indicates the strength of each edge with the highest one normalized to 1.0. 67

6.1 Minimum Effort Temporal Activity Localization (METAL): during training, we simply have untrimmed videos with only video-level labels and trimmed videos of the same label; during testing, the learned model is applied to TAL in untrimmed videos given only a few trimmed examples from unseen classes. 70
6.2 Similarity Pyramid Network (SPN) architecture for METAL under one-shot setting (best viewed in color). Both untrimmed and trimmed videos are fed into a shared Conv3D network for feature extraction, and a temporal feature pyramid is applied to summarize the untrimmed video. The features are then passed through the multi-scale relation module to obtain the similarity pyramids and similarity scores. Using these outputs, we compute two loss functions namely CSSL and PCSL, which are optimized jointly to train the network. 73
6.3 Qualitative visualization of one-shot temporal activity localization results by SPN (best viewed in color) on five different activity categories in THUMOS'14 dataset (from top to bottom): *Baseball Pitch*, *Golf Swing*, *Soccer Penalty*, *Long Jump* and *Cricket Bowling*. Ground truth activity segments are marked in blue and predicted activity segments are marked in green. 87
6.4 Qualitative visualization of one-shot temporal activity localization results by SPN (best viewed in color) on five different activity categories in ActivityNet v1.2 dataset (from top to bottom): *Washing Face*, *Removing Curlers*, *Using the Pommel Horse*, *Hand Washing Clothes* and *Vacuuming Floor*. Ground truth activity segments are marked in blue and predicted activity segments are marked in green. 88

6.5 Qualitative Visualization of the multi-scale similarity scores on four different testing batches in ActivityNet v1.2 dataset (best viewed in color) under five-way one-shot localization. The segments with top 5 similarity scores are visualized with each class in the support set shown in different colors: *red*, *orange*, *yellow*, *green* and *blue*. The predicted segments are organized into a multi-scale architecture with different temporal resolutions at each layer, and the similarity score is shown under each predicted segment. Light color indicates that the corresponding segment is suppressed by temporal NMS. For better visualization, the temporal length of each video is normalized to 1.0 and a reference time line is shown at the bottom of each example. 89

List of Tables

3.1 Temporal activity detection mAP on THUMOS'14. The top performing methods in existing papers are shown. S^3D achieves state-of-the-art performance at different overlap threshold. (- indicates that results are unavailable in the corresponding papers). 23
3.2 Effects of various design choices on S^3D performance, the span with ratio 1.0 is included by default. 24
3.3 Effects of using multiple temporal feature layers and span regression. . . 25

4.1 Activity detection results on ActivityNet v1.3 validation subset. The performances are measured by mean average precision (mAP) for different IoU thresholds and the average mAP of IoU thresholds from $0.5 : 0.05 : 0.95$. 41
4.2 Results for using a single-resolution feature map as the network input. . . 42
4.3 Results for combing multiple feature maps as the network input. 43
4.4 Results for the impact of the two-branch network architecture. 43
4.5 Results for incorporating local and global temporal contexts. 44
4.6 Comparison of our approach and the state-of-the-art methods in the approximate computation time(s) to process each video on ActivityNet dataset. 46

5.1 Natural language moment retrieval results on DiDeMo dataset. MAN outperforms previous state-of-the-art methods by $\sim 3\%$ among all metrics. 60
5.2 Natural language moment retrieval results on Charades-STA dataset. MAN significantly outperforms previous state-of-the-art methods by a large margin. 60
5.3 Ablation study for effectiveness of MAN components: Top: Advantage of a single-shot video encoder. Mid: Effectiveness of the feature alignment. Bottom: Importance of the IGAN. 61
5.4 Ablation study on different visual features. MAN with VGG-16 features already outperforms state-of-the-art method, and TAN features further boost the performance. 62

6.1 TAL results on ActivityNet v1.2 (in percentage). mAP at tIoU threshold 0.5 and average mAP are reported. Methods are categorized into three groups: Weak supervision provides video-level labels during training; Full supervision provides temporal boundary annotations during training; Few-shot refers to only a few labeled examples are available. 83
6.2 TAL results on THUMOS'14 (in percentage). mAP at tIoU threshold 0.5 is reported. The methods are categorized into the same groups as used in Table 6.1. 84
6.3 Ablation study for different SPN components on ActivityNet. Top: Weight initialization for the embedding module. Bottom: Effectiveness of temporal feature pyramid, GCN and CSSL. Results are reported under five-way one-shot localization. 85

Chapter 1

Introduction

Automatically analyzing and understanding the content of a video is one of the long-standing goals of computer vision. While deep learning has achieved near perfect accuracy in image recognition and speech processing, video understanding is still far from ideal. Existing AI systems that can interpret videos today often rely on static image analysis or can only process temporal information in a short video snippet. To automatically understand the content of long video streams, this thesis mainly describes the efforts to design accurate, efficient, and intelligent deep learning models for temporal activity detection in untrimmed videos. Detecting segments of interest from untrimmed videos is a key step towards segment-level video understanding. Depending on the purposes of tasks being performed, activity detection problems can be mainly divided to three categories: detecting activities of interest from videos without specific purposes (*i.e.*, temporal activity detection); detecting temporal segment that best corresponds to a language query (*i.e.*, natural language moment retrieval); and detecting activities given less supervision (*i.e.*, weakly-supervised or few-shot activity detection). In this dissertation, we propose novel approaches for these problems while aiming to provide potential integration among each other.

In the first work, we present a Single Shot multi-Span Detector (S^3D), a simple yet novel fully Conv3D-based framework for activity detection in continuous untrimmed video streams. While activity recognition only aims at classifying the categories of manually trimmed video clips, activity detection is substantially more challenging, as it is expected to handle activities with variable lengths, predicting both the activity category and the precise temporal boundaries of each instance. Previous works have often attempted to solve the problem using temporal proposals and separate activity classifiers, leading to low performance in accuracy and processing time. We propose a novel single-shot end-to-end model that jointly optimizes both localization of segments and recognition of activities by learning from training data. S^3D is a highly-unified network via setting multi-scale default spans at feature maps with different temporal resolutions to naturally handle activities of different lengths. The network takes as input a whole video stream, allowing our scheme to see a larger temporal context and produce better detection results. Experimental results show that our S^3D achieves state-of-the-art performance on temporal activity detection task on THUMOS'14 benchmark. Besides its strong performance, the simple S^3D network is also very efficient and can run at 1271 FPS on a single GPU.

In the second work, we investigate the multi-scale modeling problem of temporal activity detection and propose a novel Dynamic Temporal Pyramid Network (DTPN). The major obstacle that people are facing in temporal activity detection, is how to effectively model activities with various temporal length and frequency. While similar problems have been well studied in object detection, multi-scale modeling for temporal detection is still under-explored. We first identify three major challenges and propose to solve them using three novel components: (1) We sample frame sequence dynamically with different frame per seconds (FPS) to construct a natural pyramidal representation for arbitrary-length input videos. (2) We design a two-branch multi-scale temporal feature hierarchy to deal with the inherent temporal scale variation of activity instances. (3) We further exploit

the temporal context of activities by appropriately fusing multi-scale feature maps. Extensive experiments show that the proposed DTPN achieves state-of-the-art performance on the challenging ActivityNet dataset.

In the third work, we strive for natural language moment retrieval in long, untrimmed video streams to move towards real-world unconstrained activity detection. Given a verbal description, our goal is to determine the start and end time (*i.e.* localization) of the temporal segment (*i.e.* moment) that best corresponds to this given query. While this formulation opens up great opportunities for better video perception, it is substantially more challenging as it needs to model not only the characteristics of sentence and video but also their complex relations. Existing methods sample candidate moments by scanning videos with varying sliding windows, and compare the sentence with each moment individually in a multi-modal common space. Although simple and intuitive, this individualist representations of sentence and video make it hard to model semantic and structural relations among two modalities. To address the above challenges, we propose an end-to-end Moment Alignment Network (MAN) to explicitly model cross-modal and moment-wise relations in a single network. We firstly propose an Iterative Graph Adjustment Network (IGAN) adopted from Graph Convolution Network (GCN) to model relations among candidate moments in a structured graph. On the public challenging DiDeMo and Charades-STA benchmarks, MAN significantly outperforms previous state-of-the-art methods by a large margin.

In the fourth work, we conceptualize a challenging example-based temporal activity detection problem and propose a novel Similarity Pyramid Network (SPN) to tackle this problem. The success of deep learning based activity detection models heavily relies on the availability of a huge amount of labeled training data, meaning that model training requires the full annotation of the ground truth segment-level boundary for each activity instance among all possible classes, which severely limits their scalability and applicabil-

ity in real-world scenarios. To reduce the annotation efforts, we propose the Minimum Effort Temporal Activity Localization (METAL): Given only a few examples, the goal is to find the occurrences of semantically-related segments in an untrimmed video sequence while model training is only supervised by the video-level annotation. We adopt the few-shot learning technique of Relation Network and propose a novel meta-learning based framework. The main idea of our model is a similarity pyramid that directly measures partial similarities between an untrimmed video and trimmed examples at different temporal resolutions. To train the SPN with only video-level labels, we devise two complimentary loss functions to simultaneously enforce both classification and localization information. Experimental results show that our SPN achieves performance superior or competitive to state-of-the-art approaches with stronger supervision.

The remaining of this dissertation is organized as follows. We introduced the related works in Chapter 2. In Chapter 3 to Chapter 6, we detailed describe all our works for detecting activities in untrimmed videos. Chapter 7 concludes the whole dissertation.

Chapter 2

Related Work

2.1 Activity Recognition

Activity recognition is an important research topic for video analysis and has been extensively studied in the past few years. Earlier methods were often based on hand-crafted visual features. 3D motion template [2], features such as SIFT-3D [3], Action MACH [4] were used for representing temporal information for activity recognition. Later, the introduction of improved Dense Trajectory (iDT) [5, 6], feature encoding with Fisher Vector (FV) [7, 8] and VLAD [9] provided a significant boost in performance.

In the past few years, tremendous progress has been made due to the introduction of large datasets [10, 11] and the developments of deep neural networks [1, 12, 13, 14, 15, 16, 17]. Two-stream network [15] learned both spatial and temporal features by operating the network on single frames and stacked optical flows using 2D CNN such as AlexNet [18], VGG [19] and ResNet [20]. 3D CNN architecture called C3D [1] used Conv3D filters to capture both spatial and temporal information directly from raw video frames. More recently, improvements on top of the C3D architecture [14, 16, 17] as well as advanced temporal building blocks such as non-local modules [21] were proposed to

further boost the performance. However, the assumption of well-trimmed videos where the activity of interest lasts for the entire video duration limits the application of these approaches in real scenarios, where the videos are usually long and untrimmed. Although they do not consider the difficult task of localizing activity instances, these methods are widely used as the backbone network for the detection task.

2.2 Object Detection and Multi-Scale Modeling

Activity detection in untrimmed videos is closely related to object detection [22, 23, 24] in spatial images, where detection is performed by classifying region proposals into foreground classes or a background class. Earlier work [22] relied on an external region proposal method and trained a CNN classifier to classify each proposed region. Faster-RCNN [23] incorporated a region proposal network and RoI pooling to jointly generate and classify region proposals with a single network, resulting in a large improvement of the accuracy and efficiency. SSD [24] completely eliminated proposal generation and subsequent feature re-sampling stages and encapsulated all computation in a single network to directly output object locations and confidence scores. Our network is inspired by SSD [24] and adopt similar design philosophies for temporal activity detection. Like SSD [24], our S^3D model is also designed for both accuracy and efficiency in a single-shot operation.

Recognizing objects at vastly different scales is a fundamental challenge in computer vision. To alleviate the problems arising from scale variation, multi-scale pyramidal modeling forms the basis of a standard solution [25] and has been extensively studied in the spatial domain. For example, independent predictions at layers of different resolutions are used to capture objects of different sizes [26], training is performed over multiple scales [20], inference is performed on multiple scales of an image pyramid [27], feature

pyramid is directly constructed from the input image [28].

Meanwhile, the multi-scale modeling for temporal activity detection is still under-explored: Shou *et al.* [29] used a multi-scale sliding window to generate snippets of different lengths, however, such method is often inefficient during runtime due to the nature of sliding window; Zhao *et al.* [30] used temporal pyramid pooling for modeling multi-scale structures without considering complex motion dynamics, since those features were directly pooled at different levels. In Chapter 4, we provide a comprehensive study on temporal multi-scale modeling and propose an efficient end-to-end solution.

2.3 Temporal Activity Detection

Unlike activity recognition, the detection task focuses on learning how to detect activity instances in untrimmed videos with annotated temporal boundaries and instance category. The problem has recently received significant research attention due to its potential application in video data analysis.

Early approaches on activity detection mainly used temporal sliding windows as candidates and classified them with activity classifiers trained on multiple features [8, 31, 32, 33, 34]. They typically extract iDT features or pre-trained DNN features, and globally pool these features within each window to obtain the input for the SVM classifiers. However, these approaches might be computationally inefficient, because one needs to apply each activity classifier exhaustively on windows of different sizes at different temporal locations throughout the entire video.

Inspired by the success of region-based detectors in object detection [22], many recent works adopt a two-stage, proposal-plus-classification framework [35, 36, 29, 30, 37], i.e. first generating a sparse set of class-agnostic segment proposals from the input video, followed by classifying the activity categories for each proposal. A large number of these

works focus on designing better proposal schemes [35, 30, 37], while others focus on building more accurate activity classifiers [36, 29, 30]. However, most of these methods do not afford end-to-end training on either the proposal or classification stage. And the proposals are typically selected from sliding windows of predefined scales, where the boundaries are fixed and may result in imprecise localization results.

Along this line of attack, Faster-RCNN is the latest region-based object detector which is composed of end-to-end trainable proposal and classification networks, and applies region boundary regression in both stages. A few very recent works have started to apply such architecture to temporal activity detection [38, 39, 40], and demonstrated competitive detection accuracy. R-C3D [38] is a classic example that closely follows the original Faster-RCNN in many design details. Dai *et al.* [39] explicitly modeled temporal contextual information into the proposal stage. Chao *et al.* [40] proposed to use a multi-tower network with temporal contexts to further improve the detection performance. However, all these methods require a separate temporal proposal and activity classification method.

Most recently, several attempts were made towards single-shot temporal activity detection: SSAD [41] proposed to directly predict activity instances in untrimmed videos with a separate feature extraction and detection network. SS-TAD [42] have investigated the use of gated recurrent memory module in a single-stream detection framework. Our approach in Chapter 3 is one of the first within this group to propose a highly-integrated detection architecture

2.4 Natural Language Moment Retrieval

The natural language moment retrieval is a new task introduced recently [43, 44]: Given a verbal description, our goal is to determine the start and end time (*i.e.* localiza-

tion) of the temporal segment (*i.e.* moment) that best corresponds to this given query. The methods proposed in [43, 44] learn a common embedding space shared by video segment features and sentence representations and measure their similarities through sliding window [44] or handcrafted heuristics [43]. While simple and effective, these methods fail to consider the challenging alignment problems.

Until recently, several methods were proposed to closely integrate language and video representation [45, 46]: Xu *et al.* [45] proposed multilevel language and video feature fusion; TGN [46] applied frame-by-word interactions between video and language and obtained improved performance. Although these works share the same spirit with ours to better align semantic information, they fail to reason the complex cross-modal relations. In Chapter 5, our work is the first to model both semantic and structural relations together in an unified network, allowing us to directly learn the complex temporal relations in an end-to-end manner.

2.5 Visual Relations and Graph Network

Reasoning about the pairwise relationships has been proven to be very helpful in a variety of computer vision tasks [47, 48, 49, 50]. Recently, visual relations have been combined with deep neural networks in areas such as object recognition [51, 52], visual question answering [53] and action recognition [54, 55]. A variety of papers have considered modeling spatial relations in natural images [56, 57, 58], and scene graph is widely used in the image retrieval tasks [59, 60]. In the field of natural language moment retrieval: Liu *et al.* [61] proposed to parse sentence structure as a dependency tree and construct a temporal modular network accordingly; Hendricks *et al.* [62] modeled video context as a latent variable to reason about the temporal relationships. However, their reasoning relies on a hand-coded structure, thus, fail to directly learn complex temporal

relations.

Our work in Chapter 5 is inspired by the GCN [63] and other successful graph-based neural networks [64, 65]. While the original GCN is proposed to reason on a fixed graph structure, we modify the architecture to jointly optimize relations together. That is, instead of fixing the temporal relations, we learn it from the data.

2.6 Weakly Supervised Detection

Weakly supervised learning has been extensively studied for object detection [66, 67, 68]. As for activity localization, video-level label is one kind of weak supervision and has been studied in recent years. Sun *et al.* [69] was the first to consider this problem and leveraged additional supervision from web images. Hide-and-Seek [70] addressed the challenge that weakly supervised detection models usually neglect some relevant parts of the target instance. UntrimmedNet [71] proposed a framework consisting of a classification module to perform action classification and a selection module to detect important temporal segments. Most recently, AutoLoc [72] and W-TALC [73] introduced novel loss functions to further improve the performance. Although these works are trained with weak supervision, the learned models can only localize activity categories observed in the training dataset.

2.7 Few-Shot Learning

Few-shot learning refers to learning from just a few training examples per class. An increasingly popular solution for few-shot learning is meta-learning where transferable knowledge can be learned from auxiliary tasks to help with the target few-shot problem. The successful MAML approach [74] aimed to meta-learn an initial condition that is

good for fine-tuning on few-shot problems. To avoid fine-tuning, some works leverage the neural networks with memories [75, 76]. Another category of approach is metric-learning which aims to learn a set of projection functions such that when represented in this embedding, inputs are easy to recognize through similarity matching [77, 78, 79, 80]. While [77, 78] applies a fixed nearest-neighbor or linear classifier, [79] proposes to use a learnable non-linear function and demonstrates improved accuracy. Yang *et al.* [81] is the first work proposing the few-shot TAL task. It applied a sliding window approach with matching network to retrieve activity instances at each location. However, they still need the expensive boundary annotations to supervise the model training.

Our work in Chapter 6 is the first to study the METAL problem which can also be framed as a joint problem of weakly supervised TAL and few-shot TAL, while previous works only consider one aspect at a time thus cannot be applied or easily extended to tackle the more challenging METAL setting.

Chapter 3

Single Shot Multi-Span Detector via Fully 3D Convolutional Network

In this chapter, we present a novel Single Shot multi-Span Detector for temporal activity detection in long, untrimmed videos using a simple end-to-end fully three-dimensional convolutional (Conv3D) network. Our architecture, named S^3D, encodes the entire video stream and discretizes the output space of temporal activity spans into a set of default spans over different temporal locations and scales. At prediction time, S^3D predicts scores for the presence of activity categories in each default span and produces temporal adjustments relative to the span location to predict the precise activity duration. Unlike many state-of-the-art systems that require a separate proposal and classification stage, our S^3D is intrinsically simple and dedicatedly designed for single-shot, end-to-end temporal activity detection. When evaluating on THUMOS'14 detection benchmark, S^3D achieves state-of-the-art performance and is very efficient and can operate at 1271 FPS.

3.1 Introduction

Advances in deep Convolutional Neural Network (CNN) have led to significant progress in video analysis over the past few years. While the performance of activity recognition has improved a lot [5, 6, 15, 12, 13, 1], the detection performance still remains unsatisfactory [82, 83, 29]. Comparing to activity recognition, which only aims at classifying the categories of manually trimmed video clips, activity detection is for detecting and recognizing activity instances from long, untrimmed video streams. It is substantially more challenging, as it is expected to handle activities with variable lengths, predicting both the activity category and the precise temporal boundaries of each instance.

A typical framework used by many state-of-the-art systems [84, 29, 82, 36] is *detection by classification*, where temporal proposals are generated by sliding windows [84, 29] or advanced proposal methods [85, 86] and separate activity classifier is applied to predict the final detection results. However, there may be certain limitations to these frameworks: (1) Temporal proposal and classification are independent processes and optimized separately with different networks, resulting in sub-optimal performance, (2) the classification network only takes the proposal frames as input, thus forbidding it to see a larger temporal context which can be beneficial, and (3) this two-stage approach is usually slow due to inefficient proposal method and duplicate operations repeated in the proposal and classification stages.

We propose a Single Shot multi-Span Detector (S^3D), a simple yet novel fully Conv3D-based framework for activity detection in continuous untrimmed video streams. As illustrated in Figure 3.1, S^3D produces a fixed-size collection of temporal spans and scores for the presence of activity class instances in those spans, followed by a temporal non-maximum suppression step to generate the final detection results. S^3D is a highly-unified network by eliminating explicit temporal proposal and classification stages and solving

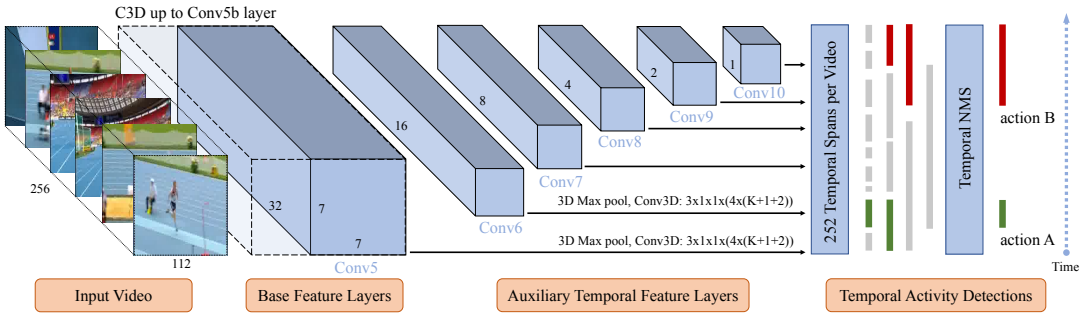

Figure 3.1: S^{3}D network architecture: Our network takes a video of 256 frames with spatial size 112×112 as input and computes base features using a standard C3D [1] network up to conv5b. We add auxiliary Conv3D layers on top of conv5 to produce a temporal feature hierarchy with multi-scale default spans at each layer. For each temporal feature map cell, we predict K class confidence scores, 1 activity confidence score and 2 location offsets with a set of Conv3D filters. Temporal NMS is applied to produce the final detection results.

the detection problem in one single shot. We set multi-scale default spans at feature maps with different temporal resolutions to naturally handle activities of different lengths. Furthermore, we predict the temporal offsets to adjust each default span in order to predict precise temporal boundaries. The network takes as input a whole video stream, allowing our scheme to see a larger temporal context and produce better detection results. The whole network is end-to-end trainable with a joint loss to directly maximize the detection performance.

The contributions are as follows: (1) We introduce S^{3}D, a single shot end-to-end activity detection model based completely on Conv3D networks that can effectively predict both the precise temporal boundaries and confidence scores of multiple activity categories in untrimmed videos. (2) We demonstrate experimentally that our S^{3}D achieves state-of-the-art performance on temporal activity detection task on THUMOS'14 benchmark. (3) Besides its strong performance, the simple S^{3}D network is also very efficient and can run at 1271 FPS on a single GPU.

3.2 Single-Shot multi-Span Detector

We introduce a Single Shot multi-Span Detector (S^3D), a simple yet novel fully Conv3D-based framework for activity detection in long untrimmed video streams. The S^3D approach, illustrated in Figure 3.1, is based on a feed-forward fully Conv3D network that produces a fixed-size collection of temporal spans and scores for the presence of activity class instances in those spans, followed by a temporal NMS step to generate the final detection results.

Our model consists of four major components: base feature layers, auxiliary temporal feature layers, multi-scale default spans and convolutional predictors. The **base feature layers** are used to extract high-level features given an input video stream. We then add **auxiliary temporal feature layers** to generate rich spatial-temporal feature hierarchies. These layers decrease in temporal dimension progressively and allow predictions of temporal spans at different locations and scales. We associate **multi-scale default spans** with each feature map cell and the default spans tile the feature map in a convolutional manner. At each feature map cell, we predict the temporal offsets relative to the default span in the cell, as well as the confidence scores that indicate the presence of an activity instance in each of those spans. These are done by adding **convolutional predictors** on top of each cell.

3.2.1 Base Feature Layers

We use Conv3D filters to extract rich feature hierarchies from a given input video stream. Specifically, the input to our model is a sequence of RGB video frames which can be represented as a tensor with dimension $\mathbb{R}^{L\times H\times W\times 3}$, where L is the number of frames, H and W are the height and width of each frame. We apply the standard C3D architecture [1] as it has been proven as an effective building block in prior works [38, 41,

42]. We adopt the Conv3D layers (conv1a to conv5b) of C3D and generate a feature map $C_{conv5} \in \mathbb{R}^{\frac{L}{8} \times \frac{H}{16} \times \frac{W}{16} \times 512}$. We use C_{conv5} as our base feature since it is a rich yet compact spatial-temporal representation of the input video stream.

3.2.2 Auxiliary Temporal Feature Layers

To allow the model to predict variable scale temporal spans, we add temporal feature layers to the end of the base feature layers. Similar to [1], we first down sample C_{conv5} by a factor of 2 in both spatial and temporal dimension via 3D max pooling and then add auxiliary Conv3D layers to produce a sequence of feature maps that progressively decrease in temporal dimension while keeping the same spatial resolution. In more detail, we stack Conv3D layers with temporal kernel size 3 to extend the temporal receptive field and the stride is set to 2 for progressively decreasing the temporal dimension. We also add bottleneck Conv3D layers to help prevent over-fitting and improve runtime efficiency. The detailed network configurations are illustrated in Figure 3.1 when $L = 256$ and $H = W = 112$.

The network is intrinsically simple by only applying Conv3D filters, but builds a rich feature hierarchy by summarizing a continuous video stream in multiple temporal resolutions, allowing us to add default temporal spans at certain layers to get temporal predictions at multiple scales.

3.2.3 Multi-scale Default Spans

To handle different activity locations and scales, [29] suggests processing the video at different segment levels and combining the results afterward, while [42] uses a gated recurrent network to assign a number of anchors at different time steps. However, by utilizing feature maps from several different layers in a single network for prediction we

can mimic the same effect, while also sharing parameters across all temporal scales. We use feature maps with different temporal resolutions for detection since earlier feature maps have higher resolution and capture finer details of the input video, and deeper feature maps have larger receptive fields and contain more temporal contexts.

In our design, we use conv5 to conv10 as our temporal feature maps and associate a set of multi-scale default spans with each temporal feature map cell. We design the tiling of default spans so that specific feature maps learn to be responsive to particular locations and lengths of the activities. Regrading a temporal feature map f with temporal length L_f, the scale of the default spans for this feature map is set as $S_f = \frac{1}{L_f}$ (as the input video length is normalized to 1). We impose different scale ratios for the default spans, and denote them as $r \in \{0.25, 0.5, 0.75, 1.0\}$. We can compute the length ($l_f^r = S_f \cdot r$) for each default span, and we set the center of each default span to $\frac{i+0.5}{L_f}$, where i indicates the i-th temporal feature cell, $i \in [0, L_f)$. So for an temporal feature map with length L_f and R different scale ratios ($R = 4$), the number of default spans is $L_f \cdot R$.

By combining predictions for all default spans with different scales from all locations of multi-scale feature maps, we have a diverse set of predictions, covering various activity locations and lengths. A concrete example is illustrated in Figure 3.2 where $L_{conv7} = 8$ and $L_{conv8} = 4$ for feature map conv7 and conv8 respectively.

3.2.4 Convolutional Predictors

Each temporal feature layer can produce a fixed set of detection predictions using a set of Conv3D filters. These are indicated on top of the feature network architecture in Figure 3.1. For a temporal feature map $C_f \in \mathbb{R}^{L_f \times H_f \times W_f \times d_f}$, the basic operation for predicting parameters of a potential temporal detection is a $3 \times H_f \times W_f$ kernel that produces scores for activity presence and categories, or temporal offsets relative to the

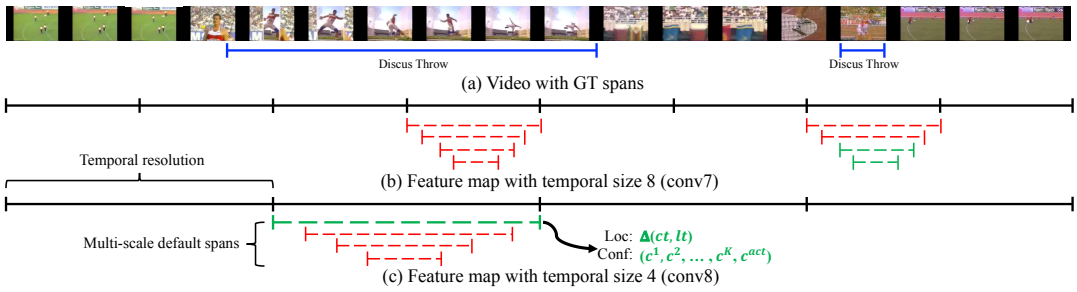

Figure 3.2: S^{3}D framework. (a) Input video with temporal ground-truth annotations. We evaluate a small set (*e.g.* 4) of multi-scale default spans at each location in several feature maps with different temporal resolutions (*e.g.* conv7 in (b) and conv8 in (c)). For each default span, we predict both the temporal offsets and the confidences for presence of activity and all activity categories. At training time, we match the default spans to the ground truth spans.

default location and scale. Specifically, for each default span at a given temporal location, we compute K positive class confidence scores plus one activity confidence score and two temporal offsets. This results in a total of $(K+1+2) \times R$ filters that are applied around each location in the feature map, yielding $(K+1+2) \times R \times L_f$ outputs for a temporal feature map C_f. For an illustration of default spans, please refer to Figure 3.2. Each default span gets a prediction score vector $v_{pred} = (c^1, c^2, ..., c^K, c^{act}, \Delta ct, \Delta lt)$ with length $K+1+2$, where c^{act} is a class-agnostic confidence score to estimate the presence of activity, c^1 to c^K are used to predict default span's category and $\Delta ct, \Delta lt$ are temporal offsets relative to the locations of default spans.

3.3 Network Training

The key step of training S^{3}D is that the ground truth information needs to be assigned to specific outputs in the fixed set of detector outputs. Once this assignment is determined, the loss function and back propagation are applied. We also discuss training data construction and hard negative mining strategies used in our model.

3.3.1 Training Data Construction and Augmentation

In theory, because S^{3}D is a fully Conv3D network, it can be applied to an input of arbitrary size. Therefore, our S^{3}D network can operate on videos of variable lengths. In practice, due to GPU memory limitations, we slide a temporal window of size L frames on the video and feed each windowed segment individually into the S^{3}D network to obtain temporal detections. Although the input window size is fixed, we decode the input video stream with a small frame rate, allowing the network to encode enough temporal contexts for precisely detecting activity instances. Therefore, given a set of training videos, we obtain a training collection of windows with temporal activity annotations inside each windowed video segment. To make the model more robust to various activity locations and scales, we further improve the training dynamics by augmenting the training videos with temporal and spatial jittering [1].

3.3.2 Matching Strategy

During training, we need to determine which default spans correspond to a ground truth detection and train the network accordingly. Specifically, for each default span, we compute the Intersection-over-Union (IoU) score with all ground truth instances. If the highest IoU score is higher than 0.5, we match the default span with the corresponding ground truth span and regard it as positive, otherwise negative. So a ground truth instance can match multiple default spans while a default span can only match one ground truth instance at most. This simplifies the learning problem, allowing the network to predict high scores for multiple overlapping default spans.

3.3.3 Hard Negative Mining

After the matching step, most of the default spans are negatives. This introduces a significant imbalance between the positive and negative training examples. Instead of using all the negative examples, we sort them using the highest activity confidence loss for each default span and pick the top ones so that the ratio between the negatives and positives is nearly $1:1$. We found that this leads to faster optimization and a more stable training.

3.3.4 Training Objective

The training objective of S^{3}D is to solve a multi-task optimization problem. Let $x_{ij}^k = \{1, 0\}$ be an indicator for matching the i-th default span to the j-th ground truth span of category $k \in [1, K]$, and s_i be the highest IoU score with any ground truth spans. The overall objective loss function is a weighted sum of the localization loss (loc), class confidence loss (conf) and activity confidence loss (act):

$$Loss = L_{loc}(x, t, g) + \alpha L_{conf}(x, c) + \beta L_{act}(s, c) \tag{3.1}$$

where α and β are the weight terms used for balancing each part of the loss function.

The localization loss is a Smooth L1 loss [87] between the predicted temporal offsets (t) and the ground truth span parameters (g). In temporal domain, we regress to offsets for the center (ct) of the default span (d) and for its length (lt):

$$L_{loc}(x, t, g) = \frac{1}{N_{pos}} \sum_{i}^{N_{pos}} \sum_{m \in \{ct, lt\}} x_{ij}^k smooth_{L1}(t_i^m - \hat{g}_j^m) \tag{3.2}$$

where N_{pos} is the number of positive matching default spans in a batch, and the tem-

poral offset parameters $\hat{g}_j^m$ are defined similarly like the bounding box offset in object detection [87]:

$$\hat{g}_j^{ct} = \Delta ct_i = (g_j^{ct} - d_i^{ct})/d_i^{lt} \qquad \hat{g}_j^{lt} = \Delta lt_i = \log(\frac{g_j^{lt}}{d_i^{lt}}) \tag{3.3}$$

where g_j^{ct}, d_i^{ct} are the centers and g_j^{lt}, d_i^{lt} are the lengths for the ground truth span and the matching default temporal span respectively.

The class confidence loss is a softmax loss over multiple class confidences (c):

$$L_{conf}(x, c) = -\frac{1}{N_{pos}} \sum_{i}^{N_{pos}} x_{ij}^k \log(\hat{c}_i^k) \tag{3.4}$$

where $\hat{c}_i^k = \frac{\exp(c_i^k)}{\sum_k \exp(c_i^k)}$ is the softmax probability for the ground truth class of this instance. The class confidence loss is only used to distinguish between multiple positive classes not including the background. We use another activity confidence score to predict activity class agnostic scores.

The activity confidence loss is a binary classification loss using sigmoid cross-entropy. Rather than using a hard ground truth score for positive (1) and negative (0), we use the IoU score s_i as ground truth for each default span. This helps the training procedure since positive default spans are assigned different confidence levels based on its overlap with the ground truth span. We define the activity confidence loss as:

$$L_{act}(s, c) = -\frac{1}{N} \sum_{i}^{N} (s_i \log(c_i^{act}) + (1 - s_i) \log(1 - c_i^{act})) \tag{3.5}$$

where N is the number of total training default spans in a batch and $N = N_{pos} + N_{neg}$; c_i^{act} is the predicted activity confidence score. Note that we separate the activity confidence score and class confidence scores via two separate losses. Comparing to only having one

softmax classification loss containing all positive classes and one background class, we find this configuration is more robust, leads to better validation performance and makes the network architecture more flexible.

3.3.5 Prediction

Activity prediction in S^3D is single shot with one forward pass of the network. Given an input video stream, we generate all default spans with class confidence scores, activity confidence score and temporal location offsets. The temporal location offset is in the form of relative displacement of the center point and length of each instance as described in Equation 3.3, which is applied on the default span to predict accurate start time and end time. Then the default spans with low activity confidence score will be filtered out and the remaining spans are refined via NMS with threshold value 0.5. Each remaining span is considered as a positive prediction and assigned the activity label with the highest class confidence score, which we consider as the final temporal detection results of S^3D.

3.4 Experiments

We evaluate the proposed framework on the THUMOS'14 [11] large-scale activity detection benchmark dataset. As shown in the experiments, our S^3D not only achieves state-of-the-art performance but also acquires fast runtime speed at 1271 FPS.

Dataset [11]. The temporal activity detection task of THUMOS'14 dataset is challenging and widely used. Over 20 hours of video and 20 activity categories are involved and annotated temporally, resulting in 200 validation and 213 test untrimmed videos. Following the standard practice, we train our models on the validation set and evaluate them on the testing set. We follow the conventional metrics used in THUMOS'14, computing the Average Precision (AP) for each activity category and calculating mean AP (mAP)

IoU threshold	0.3	0.4	0.5	0.6	0.7
S-CNN [29]	36.3	28.7	19.0	10.3	5.3
CDC [36]	40.1	29.4	23.3	13.1	7.9
SSAD [41]	43.0	35.0	24.6	-	-
TCN [39]	-	33.3	25.6	15.9	9.0
R-C3D [38]	44.8	35.6	28.9	-	-
SSN [30]	**50.6**	40.8	29.1	-	-
SS-TAD [42]	40.1	-	29.2	-	9.6
S^3D	47.9	**41.2**	**32.6**	**23.3**	**14.3**

Table 3.1: Temporal activity detection mAP on THUMOS'14. The top performing methods in existing papers are shown. S^3D achieves state-of-the-art performance at different overlap threshold. (- indicates that results are unavailable in the corresponding papers).

for evaluation.

Implementation Details. S^3D takes as input $L = 256$ raw video frames with size $H = W = 112$. We decode each video at 8 FPS and produce a collection of training windows. Thus, each window contains 32 seconds of a video stream and this is motivated by the fact that more than 99% of activity instances in THUMOS'14 are less than 32 seconds. We use conv5 to conv10 as the temporal feature layers with temporal dimension $\{32, 16, 8, 4, 2, 1\}$ and associate a set of default spans at each temporal feature cell with four ratios $\{0.25, 0.5, 0.75, 1.0\}$, resulting in 252 default spans in total; the default spans correspond to spans of duration between 0.25s and 32s uniformly distributed at different temporal locations. We initialize base feature layers with C3D weights pre-trained on Sports-1M by the authors in [1], and other layers from scratch. We allow all the layers of S^3D to be trained on THUMOS'14 with the end-to-end loss function.

3.4.1 Comparison with State-of-the-art

The comparison results between our S^3D and other top-performing methods are summarized in Table 3.1, and our S^3D outperforms all previous state-of-the-art methods.

Furthermore, S^{3}D improves the state-of-the-art by a large margin when the evaluation IoU thresholds are set at higher levels (0.5 to 0.7), indicating its superior ability to predict precise temporal boundaries of different activities.

In comparison with the proposed S^{3}D model: previous systems on top of C3D networks (S-CNN [29], CDC [36]) largely relies on good temporal proposals generated by external proposal methods, restricting them from directly optimizing the detection performance. R-C3D [38] is able to process a long video stream and predict multi-scale activity instances, but it only applies anchors on a single feature map with fixed temporal dimension. With the proposed S^{3}D framework, we jointly optimize the feature representation and detection layers at different temporal levels by processing an untrimmed input video stream with enough temporal context.

3.4.2 Ablation Study

To understand S^{3}D better, we evaluate our network with different variants on THUMOS'14 to study their effects. For all experiments, we only change the certain part of the network and use the same evaluation settings. We compare the result of different variants using the mAP at IoU threshold 0.5.

include 1.0 span	✓	✓	✓	✓
include 0.25 span		✓	✓	✓
include 0.5 span			✓	✓
include 0.75 span				✓
# Spans	63	126	189	252
mAP@0.5	27.5	29.5	31.1	**32.6**

Table 3.2: Effects of various design choices on S^{3}D performance, the span with ratio 1.0 is included by default.

Default Span Ratio. By default, we use 4 default spans per each temporal location. If we remove the spans with ratio 0.75, the mAP drops by 1.5%. By further removing the

spans with ratio 0.25 and 0.5, the mAP drops another 3.6%. By only keeping the span with ratio 1.0, our model already has a strong performance (mAP 27.5%) since it already covers most ground truth instances in the dataset. Using a variety of default ratios make the task of predicting spans easier for the network and result in better performance.

Span Regression. The default spans are defined at fixed temporal locations. In order to generate precise predictions for starting and ending time of each activity instance, we adjust each default span by applying a temporal offset described in Equation 3.3. This technique, which we call span regression, allows our model to predict temporal spans at much smaller granularities. As shown in Table 3.3, span regression improves the mAP from 28.6% to 32.6%.

Span regression	conv5	conv6	conv7	conv8	conv9	conv10	mAP@0.5	# Spans
✓	✓	✓	✓	✓	✓	✓	**32.6**	252
	✓	✓	✓	✓	✓	✓	28.6	252
✓	✓	✓	✓	✓	✓		31.8	248
✓	✓	✓	✓	✓			30.7	240
✓	✓	✓	✓				27.6	224

Table 3.3: Effects of using multiple temporal feature layers and span regression.

Multi-scale Default Spans. A major advantage of S^3D is using default spans of different scales on different temporal feature layers. To measure the advantage gained, we progressively remove layers and compare results. Table 3.3 shows a decrease in accuracy with fewer layers, dropping monotonically from 32.6% to 27.6%. This is because that different layers are responsible for predicting temporal activities at different lengths, which reinforces the message that it is critical to spread spans of different scales over different layers.

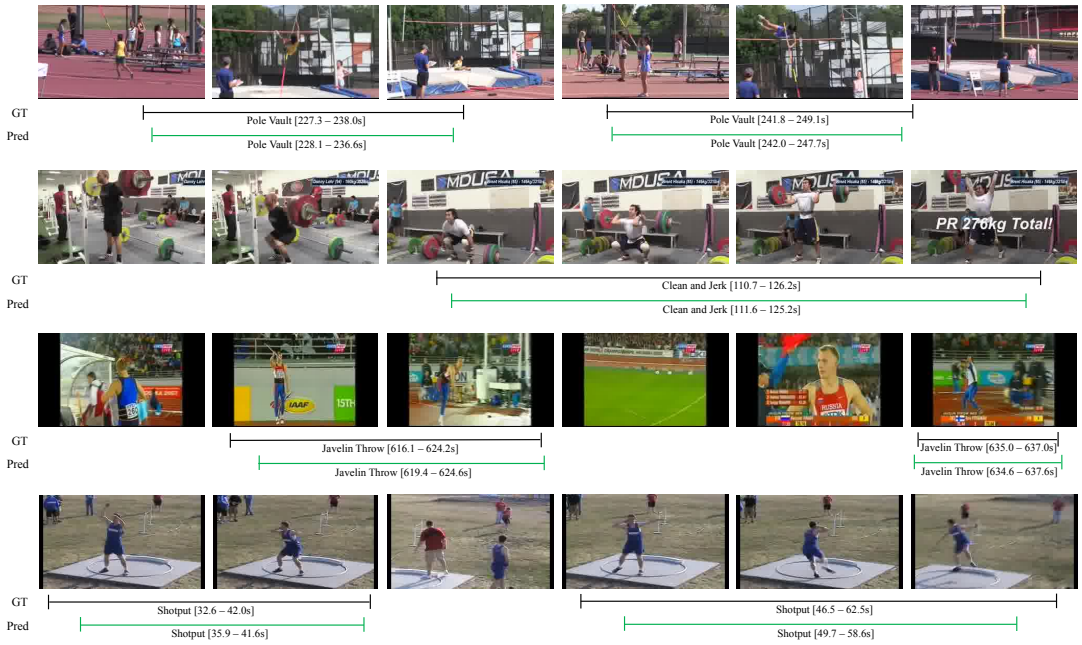

Figure 3.3: Qualitative visualization of the top detected activities by S^{3}D (best viewed in color) on four different activity categories in THUMOS'14 dataset: *Pole Vault*, *Clean and Jerk*, *Javelin Throw* and *Shotput*. Ground truth activity segments are marked in black and predicted activity segments are marked in green.

3.4.3 Qualitative Results

We provide qualitative results to demonstrate the effectiveness and robustness of our proposed S^{3}D network. As shown in Figure 3.3, different video streams contain very diversified background context and different activity instances vary a lot in temporal location and scale. S^{3}D is able to predict the accurate temporal span as well as the correct activity category. Furthermore, S^{3}D can distinguish activity with minor differences such as the normal weightlifting compared to *Clean and Jerk*. It is also capable of detecting the same activity sequence with different playing speed as shown in the *Shotput* example.

Since our model has a single-shot, end-to-end design with simple Conv3D building blocks, it is also very efficient. We benchmark our model on a GeForce GTX 1080 Ti GPU, and our S^{3}D can run much faster than real time at 1271 FPS. For comparison,

previous top performing methods [29, 41, 36] have significantly lower FPS for the whole detection pipeline. Comparing to some recent works [38, 42] providing good runtime efficiency, our S^{3}D achieves much better accuracy.

3.5 Conclusion

In this chapter, we introduce S^{3}D, a Single Shot multi-Span Detector for temporal activity detection. We design a simple network architecture by using only a fully Conv3D network on top of the raw video frames to jointly predict the temporal boundaries as well as activity categories. A key feature of S^{3}D is the use of multi-scale temporal span outputs attached to multiple temporal feature maps. With this framework, we achieved state-of-the-art performance on THUMOS'14 benchmark dataset, while being efficient to run much faster than real time on a single GPU.

Chapter 4

Dynamic Temporal Pyramid Network for Temporal Multi-Scale Modeling

Recognizing instances at varying scales simultaneously is a fundamental challenge in visual detection problems. While spatial multi-scale modeling has been well studied in object detection, how to effectively apply a multi-scale architecture to temporal models for activity detection is still under-explored. In this chapter, we identify three unique challenges that need to be specifically handled for temporal activity detection. To address all these issues, we propose Dynamic Temporal Pyramid Network (DTPN), a new activity detection framework with a multi-scale pyramidal architecture featuring three novel designs: (1) We sample frame sequence dynamically with different frame per seconds (FPS) to construct a natural pyramidal representation for arbitrary-length input videos. (2) We design a two-branch multi-scale temporal feature hierarchy to deal with the inherent temporal scale variation of activity instances. (3) We further exploit the temporal context of activities by appropriately fusing multi-scale feature maps, and demonstrate

that both local and global temporal contexts are important. By combining all these components into a uniform network, we end up with a single-shot activity detector involving single-pass inferencing and end-to-end training. Extensive experiments show that the proposed DTPN achieves state-of-the-art performance on the challenging ActvityNet dataset.

4.1 Introduction

Temporal activity detection has drawn increasing interests in both academic and industry communities due to its vast potential applications in security surveillance, behavior analytics, videography and so on. One major obstacle that people are facing in temporal activity detection, is how to effectively model activities with various temporal length and frequency. Especially, the challenge of localizing precise temporal boundaries among activities of varying scales has been demonstrated as one major factor behind the difference in performance [39]. Luckily, the problem of scale variation is not new in computer vision researches, as it has been well studied in object detection in images [88]. In order to alleviate the problems arising from scale variation and successfully detect objects at multiple scales, extensive analysis has been conducted in recent years. Multi-scale pyramidal architecture has been widely adopted and become a general structure in many state-of-the-art object detection frameworks [24, 28].

How to effectively model the temporal structure for activity detection using a multi-scale pyramidal network then? To answer this question, we first identify three unique problems that need to be specifically handled for temporal activity detection: (1) The duration of the input video is arbitrary (usually ranges from few seconds to few minutes). A naive subsampling method (resize the video) or sliding window (crop the video) will fail to fully exploit the temporal relations. (2) The temporal extent of activities varies

dramatically compared to the size of objects in an image, posing a challenge to deal with large instance scale variation. (3) The spatial context of a bounding box is important to correctly classify and localize an object, and the temporal context is arguably more so than the spatial context. Thus, cross-scale analysis becomes much more crucial in temporal domain. In this work, we propose a multi-scale pyramidal deep-learning architecture with three novel elements designed to solve the above problems accordingly.

1. **How to effectively extract a feature representation for input video of arbitrary length?** A common practice in most existing works [89, 40, 39] is to use a high-quality video classification network for extracting a feature representation from raw frame sequence. However, when dealing with input video of arbitrary length, they only decode the video at a fixed FPS and extract features with a single resolution. To fully exploit temporal relations at multiple scales and effectively construct a feature representation, we propose to use dynamic sampling to decode the video at varying frame rates and construct a pyramidal feature representation. Thus, we are able to parse an input video of arbitrary length into a fixed-size feature pyramid without losing short-range and long-range temporal structures. Nevertheless, our extraction method is very general and can be applied to any framework and compatible with a wide range of network architectures.

2. **How to build better temporal modeling architectures for activity detection?** In dealing with the large instance scale variation, we draw inspirations from SSD [24] to build a multi-scale feature hierarchy allowing predictions at different scales by appropriately assigning default spans. This multi-scale architecture enforces the alignment between the temporal scope of the feature and the duration of the default span. Besides, we also draw inspirations from Faster-RCNN [23] to use separate features for classification and localization since features for localiza-

tion should be sensitive to pose variation while those for classification should not. We propose a new architecture to leverage the efficiency and accuracy from both frameworks while still maintaining a single shot design. In our work, we use separate temporal convolution and temporal pooling branches with matched temporal dimension at each scale, and use a late fusion scheme for final prediction.

3. **How to utilize local and global temporal contexts?** We claim both local temporal context (*i.e.*, moments immediately preceding and following an activity) and global temporal context (*i.e.*, what happens during the whole video duration) are crucial. We propose to explicitly encode local and global temporal contexts by fusing features at appropriate scales in the feature hierarchy.

Our contributions are: (1) We take a closer look at multi-scale modeling for temporal activity detection and identify three unique challenges. (2) To address all these issues in a single network, we introduce the Dynamic Temporal Pyramid Network (DTPN), which is a single shot activity detector featuring a novel multi-scale pyramidal architecture design. (3) Our DTPN achieves state-of-the-art performance on temporal activity detection task on ActivityNet benchmark [10].

4.2 Dynamic Temporal Pyramid Network

We present a *Dynamic Temporal Pyramid Network (DTPN)*, a novel approach for temporal activity detection in long untrimmed videos. DTPN is dedicatedly designed to address the temporal modeling challenges as discussed in the introduction with a multi-scale pyramidal architecture. The overall DTPN framework is a single-shot, end-to-end activity detector featuring three novel architectural designs: pyramidal input feature extraction with dynamic sampling, multi-scale feature hierarchy with two-branch

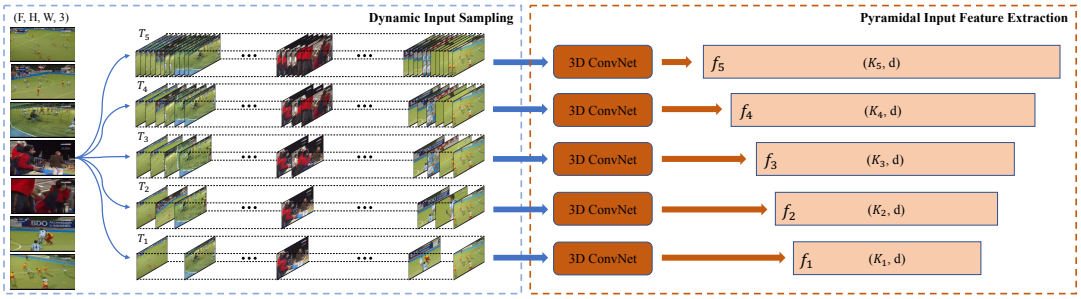

Figure 4.1: An illustration of pyramidal input feature extraction with 5 sampling rates. Left: input video is sampled at different FPS to capture motion dynamics at different temporal resolutions; Right: a shared 3D ConvNet is used to extract the input feature at each resolution.

network, and local and global temporal context.

4.2.1 Pyramidal Input Feature Extraction with Dynamic Sampling

The input of our network is an untrimmed video with an arbitrary length. We denote a video ν as a series of RGB frames $\nu = \{I_i\}_{i=1}^{F}$, where $I_i \in \mathbb{R}^{H \times W \times 3}$ is the i-th input frame and F is the total number of frames. A common practice is to use a high-quality video classification network to extract a 1D feature representation on top of the input frame sequence [89, 40, 90]. This feature extraction step is beneficial for summarizing spatial-temporal patterns from raw videos into high-level semantics. The backbone classification network can be of any typical architectures, including the two-stream network [15], C3D [1], I3D [16], Res3D [14], P3D [17], etc. However, an obvious problem of the classification ConvNet in their current form is their inability in modeling long-range temporal structure. This is mainly due to their limited temporal receptive field as they are designed to operate only on a single stack of frames in a short snippet.

To tackle this issue, we propose to extract pyramidal input feature with dynamic sampling, a video-level framework to model multi-level dynamics throughout the whole video.

Sparse sampling has already been proven very successful when solving the video classification problem [91], where preliminary prediction results from short snippets sparsely sampled from the video are aggregated to generate the video-level prediction. Following similar ideas, we propose a general feature extraction framework specifically for the temporal activity detection.

Formally, given an input video ν with F frames and a sampling scale index s, we divide the entire frame sequence into K_s different segments of equal duration. Suppose a classification network takes w frames as input and generates a d-dimensional 1D feature vector before any classification layers, we uniformly sample w frames in each segment to construct a sequence of snippets $\{T_1, T_2, ..., T_{K_s}\}$, where each $T_i, i \in [1, K_s]$ is a snippet of w frames which can be directly used as an input to the backbone network. Thus, we can extract features for a specific sampling scale index s as

$$f_s = \bigcup_{i=1}^{K_s} F(T_i, \mathbf{W}) \in \mathbb{R}^{K_s \times d} \tag{4.1}$$

where $F(T_i, \mathbf{W})$ is the function representing a ConvNet with parameter $\mathbf{W}$ which operates on snippet T_i and generates a d-dimensional feature vector. Thus, each single feature vector $F(T_i, \mathbf{W})$ in f_s covers a temporal span of $\frac{F}{K_s}$ frames. Suppose the input frame sequence is decoded at r FPS, then the equivalent feature-level sampling rate is given as $\frac{r \times K_s}{F}$. Instead of only extracting features at a single scale, we apply a set of different scales to construct a pyramidal input feature, which can be considered as sampling the input frame sequence with dynamic FPS. Technically, we use S different scales to sample the input video with a base scale length K_1 and an up sampling factor of 2. *i.e.* $K_s = 2^{s-1} \times K_1, s \in [1, S]$ different feature vectors will be extracted given a scale index s. This dynamic sampling procedure allows us to directly summarize both short-range and long-range temporal relations while being efficient during runtime. Finally, a pyramidal

feature is constructed as

$$f_{pymd} = \bigcup_{s=1}^{S} f_s, f_s \in \mathbb{R}^{K_s \times d} \tag{4.2}$$

which will be used as the input to the two-branch network (Sec. 4.2.2).

The overall procedure is illustrated in Fig. 4.1. Note that our approach is different from temporal pyramid pooling [30] where higher-level features are directly pooled, and multi-scale sliding window [29] where a window size is pre-defined. Our dynamic sampling approach fixes the number of sampling windows and computes independent features by directly looking at input frames with different receptive fields. We find that both sparse and dense sampling are important for temporal detection task: sparse sampling is able to model long-range temporal relations. Dense sampling, on the other hand, provides high-resolution short-range temporal features. By using an off-the-shelf video classification network and a dynamic frame sampling strategy, we are able to construct a pyramidal input feature that naturally encodes the video at varying temporal resolutions.

Comparison with previous works. When extracting features from the input video, previous works [90, 39, 40, 89, 92] decode the input video with a fixed FPS (usually small for computational efficiency) and extract features using a non-overlapping sliding window, which corresponds to a fixed FPS single-scale sampling in our schema. Although advanced networks are applied to model temporal relationships, their feature extraction component fails to fully exploit the multi-scale motion context in an input video stream. More importantly, our extraction strategy is very general thus can be applied to any framework and compatible with a wide range of network architectures.

4.2.2 Multi-scale Feature Hierarchy with Two-branch Network

To allow the model to predict variable scale temporal spans, we follow the design of SSD to build a multi-scale feature hierarchy consisting of feature maps at several scales

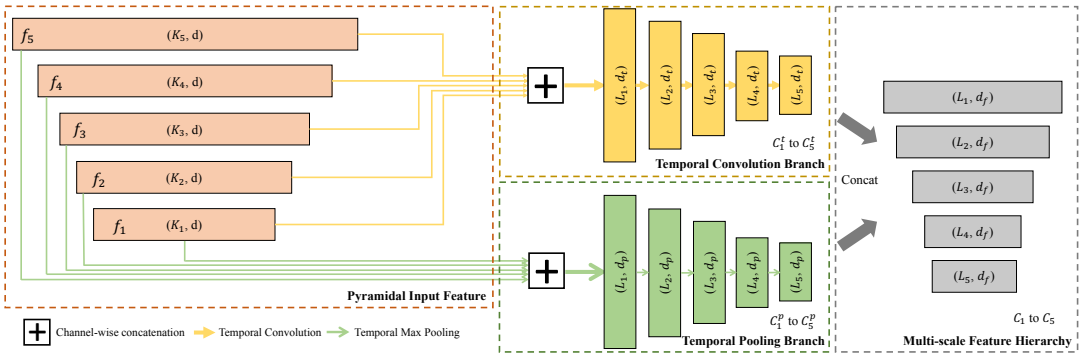

Figure 4.2: An illustration of the two-branch multi-scale network with $S = N = 5$. The network combines a temporal convolution branch and a temporal pooling branch, where the features are concatenated and down sampled. Late fusion scheme is applied to build the multi-scale feature hierarchy.

with a scaling step of 2. We then assign default temporal spans at each layer to get temporal predictions at multiple scales. More specifically, a multi-scale feature hierarchy is created which we denote as $\{C_i\}_{i=1}^{N}, C_i \in \mathbb{R}^{L_i \times d_f}$ where N is the total number of features each with a temporal dimension L_i and feature dimension d_f. For a simple and efficient design, we set $L_1 = K_1$ and $L_N = 1$, and the temporal dimension in between follows $L_i = 2L_{i+1}$.

The next question is: how do we combine the pyramidal input feature and build the multi-scale network? As illustrated in Fig. 4.2, we propose to use a two-branch network, *i.e.*, a temporal convolution branch and a temporal pooling branch to fuse the pyramidal input feature and aggregate these branches at the end. This design choice is inspired by the fact that pooling features contain more translation-invariant semantic information which is classification-friendly and convolutional features better model temporal dynamics which are helpful for localization [23, 28].

In more detail, both branches take as input the pyramidal feature f_{pymd}. For the temporal convolution branch, a Conv1D layer with temporal kernel size $\frac{K_s}{L_1}+1$, stride $\frac{K_s}{L_1}$ is applied to each input feature $f_s \in f_{pymd}, s \in [1, S]$ to increase the temporal receptive field

and decrease the temporal dimension to L_1 (temporal stride is set to 1 for f_1 since no down sampling is needed). We use channel-wise concatenation to combine the resulting features into a single feature map $C_1^t \in \mathbb{R}^{L_1 \times d_t}$. Based on C_1^t, we stack Conv1D layers with kernel size 3 and stride 2 for progressively decreasing the temporal dimension by a factor 2 to construct C_2^t through C_N^t. For the temporal pooling branch, a non-overlapping temporal max pooling with window size $\frac{K_s}{L_1}$ is used on top of each input feature $f_s \in f_{pymd}, s \in [1, S]$ to match with the temporal dimension L_1. Similar to the temporal convolution branch, channel-wise concatenation is applied here to construct $C_1^p \in \mathbb{R}^{L_1 \times d_p}$. Then, we use temporal max pooling with a scaling step of 2 to construct the feature hierarchy $\{C_i^p\}_{i=1}^N$. Finally, features from the two branches are aggregated together to generate the final feature hierarchy $\{C_i\}_{i=1}^N$, which will be used to further model the temporal context (Sec. 4.2.3).

Simplicity is central to our design and we have found that our model is robust to many design choices. We have experimented with other feature fusion blocks such as element-wise product, average pooling, etc., and more enhanced building blocks such as dilated convolution [93] and observed marginally better results. Designing better network blocks is not the focus of this work, so we opt for the simple design described above.

Comparison with previous works. Previous works based on SSD framework [90, 92] only use a single convolutional branch and don't apply feature fusion since only a single-scale input is applied. Our design uses two separate branches with slightly different feature designs at multiple scales. The localization branch uses temporal convolution for better localization while the classification branch uses maximum pooling to record the most prominent features for recognition. We show experimentally that our two-branch design achieves much better results compared to single-branch (Sec. 4.3.3).

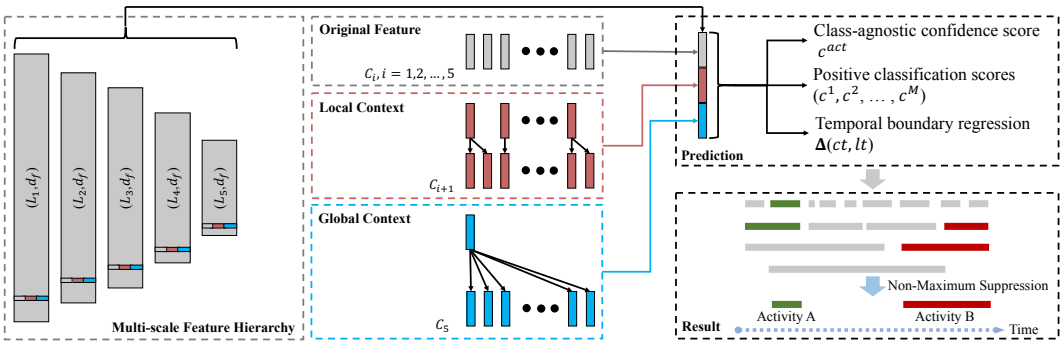

Figure 4.3: An illustration of local and global contexts when setting N to 5. Every temporal feature map cell at all scales is enhanced by its local context (next scale) and global context (last scale) to produce a set of prediction parameters. Temporal NMS is applied to produce the final detection results.

4.2.3 Local and Global Temporal Contexts

Temporal contextual information has been shown to be critical for temporal activity detection [39, 40]. There are mainly two reasons: First, it enables more precise localization of temporal boundaries. Second, it provides strong semantic cues for identifying the activity class. In order to fully utilize the temporal contextual information, we propose to use both local temporal context (*i.e.*, what happens immediately before and after an activity instance) and global temporal context (*i.e.*, what happens during the whole video duration). Both contexts help with localization and classification subtasks but with different focuses: local context focuses more on localization with immediate cues to guide temporal regression, while global context tends to look much wider at the whole video to provide classification guidance. Below, we detail our approach.

Our multi-scale feature hierarchy can easily incorporate contextual information since it naturally summarizes temporal information at different scales. To exploit the local temporal context for a specific layer C_i, we combine each temporal feature cell at C_i with a corresponding feature cell at C_{i+1}. Specifically, we first duplicate each feature cell at C_{i+1} twice to match with the temporal dimension of C_i and concatenate the feature

maps together. Thus, at each feature cell location in C_i, it not only contains feature at its original location but also a local context feature at the next scale. To exploit the global temporal context, instead of looking at the feature map in the next scale, we combine the feature with the last feature map C_N which summarizes the whole video content. Similar to the local temporal context, we duplicate C_N to have the same temporal dimension with C_i. We exploit local and global contexts at all layers in our network, thus, each temporal feature cell is enhanced by its local and global temporal information. We illustrate this mechanism in Fig. 4.3.

Each temporal feature map can produce a fixed set of detection predictions using a set of Conv1D layers. These are indicated on top of the feature network in Fig. 4.3. The basic operation for predicting parameters of a potential temporal detection is a Conv1D filter that produces scores for activity presence (c^{act}) and categories (c^1 to c^M, where M is the total number of classes), and temporal offsets ($\Delta ct, \Delta lt$) relative to the default temporal location. The temporal detections at all scales are combined through temporal non-maximum suppression for generating the final detection results.

Comparison with previous works. Neither Zhang *et al.* [92] nor Lin *et al.* [90] exploited any context features in their network. Dai *et al.* [39] included context features in the proposal stage, but they pooled features from different scales. Chao *et al.* [40] only exploited the local temporal context. Our work considers both local and global temporal contexts and inherently extract contexts from a multi-scale temporal feature hierarchy.

4.3 Experiments

We evaluate the proposed framework on the ActivityNet [10] large-scale temporal activity detection benchmark. As shown in the experiments, our DTPN achieves state-of-the-art performance. We also perform a set of ablation studies to analyze the impact

of different components in our network.

Dataset. ActivityNet [10] is a recently released dataset which contains 200 different types of activities and a total of 849 hours of videos collected from YouTube. ActivityNet is the largest benchmark for temporal activity detection to date in terms of both the number of activity categories and number of videos, making the task particularly challenging. There are two versions, and we use the latest version 1.3 which contains 19994 untrimmed videos in total and is divided into three disjoint subsets, training, validation, and testing by a ratio of $2:1:1$. On average, each activity category has 137 untrimmed videos. Each video on average has 1.41 activities which are annotated with temporal boundaries. Since the ground-truth annotations of test videos are not public, following traditional evaluation practices on this dataset, we use the validation set for ablation studies.

Evaluation Metrics. ActivityNet dataset has its own convention of reporting performance metrics. We follow their conventions, reporting mean average precision (mAP) at different IoU thresholds 0.5, 0.75 and 0.95. The average of mAP values with IoU thresholds $[0.5:0.05:0.95]$ is used to compare the performance between different methods.

4.3.1 Implementation Details

Feature Extractor. To extract the feature maps, we first train a Residual 3D ConvNet (Res3D) model [14] on the Kinetics activity classification dataset [16]. The model takes as input a stack of 8 RGB frames with spatial size 256×256, performs 3D convolutions, and extracts a feature vector with $d = 2048$ as the output of an average pooling layer. We decode each video at 30 FPS to take enough temporal information into account, and each frame is resized to 256×256. We set $K_1 = L_1 = 16$ and $S = 5$ for dynamic sampling, thus, we divide the input frame sequence into a set of $\{16, 32, 64, 128, 256\}$ segments and

a snippet of window size $w = 8$ is sampled in each segment. Each snippet is then fed into our Res3D model to extract a pyramidal input feature. Note that feature extraction can be done very efficiently with a single forward pass in batches.

Temporal Anchors. In our design, we associate a set of temporal anchors with each temporal feature map cell in the multi-scale feature hierarchy $\{C_i\}_{i=1}^{5}$. As described in Sec. 4.2.2, the temporal dimension of C_i is given as $L_i = 2^{5-i}, i \in [1, 5]$. Regarding a feature map C_i, we set the length of each temporal anchor to be $\frac{1}{L_i}$ (as the input video length is normalized to 1), and the centers are uniformly distributed with a temporal interval of $\frac{1}{L_i}$ in between. Thus, we assign a set of $\{16, 8, 4, 2, 1\}$ temporal anchors in our network which correspond to anchors of duration between $\frac{1}{16}$ and the whole video length. This allows us to detect activity instances with varying scales.

Network Configurations. Our system is implemented in TensorFlow [94]. All evaluation experiments are performed on a work station with NVIDIA GTX 1080 Ti GPUs. For multi-scale feature hierarchy, we generate a set of features with temporal dimension $\{16, 8, 4, 2, 1\}$ through both temporal convolution branch and temporal pooling branch as described in Sec. 4.2.2. In temporal convolution branch, we set the number of filters to 64 for five different input features, and $d_t = 320$ for all convolutional layers after concatenation. When training the network, we randomly flip the pyramidal input feature along temporal dimension to further augment the training data. The network is trained with multi-task end-to-end loss functions involving a regression loss, a classification loss and a localization loss. The whole network is trained for 20 epochs with the learning rate set to 10^{-4} for the first 12 epochs and 10^{-5} for the last 8 epochs.

IoU threshold	0.5	0.75	0.95	Average
Singh and Cuzzolin [95] (2016)	34.47	-	-	-
Wang and Tao [96] (2016)	45.10	4.10	0.00	16.40
Shou *et al.* [97] (2017)	45.30	26.00	0.20	23.80
Xu *et al.* [89] (2017)	26.80	-	-	12.70
Dai *et al.* [39] (2017)	36.44	21.15	**3.90**	-
Chao *et al.* [40] (2018)	38.23	18.30	1.30	20.22
DTPN (ours)	**41.44**	**25.49**	3.26	**25.72**

Table 4.1: Activity detection results on ActivityNet v1.3 validation subset. The performances are measured by mean average precision (mAP) for different IoU thresholds and the average mAP of IoU thresholds from $0.5 : 0.05 : 0.95$.

4.3.2 Comparison with State-of-the-art

Table 4.1 shows our activity detection results on the ActivityNet v1.3 validation subset along with state-of-the-art methods [95, 96, 97, 89, 39, 40] published recently. The proposed framework, using a single model instead of an ensemble, is able to achieve an average mAP of 25.72 that tops all other methods and perform well at high IoU thresholds, *i.e.*, 0.75 and 0.95. This clearly demonstrates the superiority of our method.

Note that the top half in Table 4.1 are top entries for challenge submission: our method is worse than [96] at IoU threshold 0.5 but their method is optimized for 0.5 overlap and its performance degrades significantly at high IoU thresholds, while our method achieves much better results (25.49 vs. 4.10 at IoU threshold 0.75); Shou *et al.* [97] builds a refinement network based on the result of [96], although they are able to improve the accuracy our method is still better when measured by the average mAP (25.72 vs. 23.80). We believe the performance gain comes from our advanced temporal modeling design for both feature extraction and feature fusion, as well as rich temporal contextual information.

IoU threshold	0.5	0.75	0.95	Average
Single-256	36.75	22.09	1.94	22.18
Single-128	36.93	21.93	2.86	22.32
Single-64	35.47	21.39	2.56	21.63
Single-32	35.62	21.78	2.57	21.66
Single-16	33.64	20.69	1.82	20.63
Pyramidal Input	**38.89**	**23.82**	**3.25**	**24.07**

Table 4.2: Results for using a single-resolution feature map as the network input.

4.3.3 Ablation Study

To understand DTPN better, we evaluate our network with different variants on ActivityNet dataset to study their effects. For all experiments, we only change a certain part of our model and use the same evaluation settings. We compare the result of different variants using the mAP at 0.5, 0.75, 0.95 and the average mAP. For a fair comparison, we don't concatenate contextual features in all experiments unless explicitly noted.

Dynamic Sampling vs. Single-resolution Sampling. A major contribution of DTPN is using dynamic sampling to extract a pyramidal input feature as the network input. However, as a general SSD based temporal activity detector, single-resolution feature can also be applied as the input to our network. We validate the design for dynamic sampling pyramidal input by comparing with single-resolution sampling input: we keep the multi-scale feature network with 5 temporal dimensions from 16 to 1 and the two-branch architecture, but instead of taking the pyramidal feature as input we only input a separate feature map of temporal size 256, 128, 64, 32 and 16 independently. The hidden dimension for each layer is kept the same for a fair comparison. The results are reported in Table 4.2. Pyramidal input performs uniformly the best compared to single input, despite the network design, this clearly demonstrates the importance of multi-scale pyramidal feature extraction.

Multi-scale Feature Fusion. We further validate our design to combine multiple

256	128	64	32	16	Average mAP
✓	✓				22.52
			✓	✓	22.01
✓		✓		✓	23.11
✓	✓	✓	✓	✓	**24.07**

Table 4.3: Results for combing multiple feature maps as the network input.

IoU threshold	0.5	0.75	0.95	Average
TConv	27.12	14.70	1.34	15.12
TPool	29.77	17.24	2.16	17.12
TConv+TPool (two-branch)	**38.89**	**23.82**	**3.25**	**24.07**

Table 4.4: Results for the impact of the two-branch network architecture.

features as our network input. Instead of just using a single-resolution feature as input, we investigate the effects of combining different input features. We also keep the same hidden dimension for each layer for a fair comparison. Table 4.3 compares different combination schemes: we observe that only dense sampling (256+128) or sparse sampling (32+16) leads to inferior performance compared to sampling both densely and sparsely (256+64+16); By adding more fine-grained details (128 and 32), our pyramidal input achieves the best result.

Two-branch vs. Single-branch. Here, we evaluate the impact of the two-branch network architecture. In our design, We propose to use a separate temporal convolution branch and temporal pooling branch and fuse the two feature hierarchies at the end. However, either branch can be used independently to predict the final detection results. Table 4.4 lists the performance of models with temporal convolution branch only (TConv) and temporal pooling branch only (TPool). We conclude that two-branch architecture can significantly improve the detection performance (more than 5% in comparison with single-branch).

IoU threshold	0.5	0.75	0.95	Average
w/o Context	38.89	23.82	3.25	24.07
w/ Local Context	40.01	24.50	3.24	24.70
w/ Global Context	40.17	24.20	**3.54**	24.62
w/ Local+Global Contexts	**41.44**	**25.49**	3.26	**25.72**

Table 4.5: Results for incorporating local and global temporal contexts.

Local and Global Temporal Contexts. We contend that temporal contexts both locally and globally are crucial for temporal activity detection. Since local and global contextual features are extracted from different layers and combined through concatenation, we can easily separate each component and see its effect. As reported in Table 4.5, We compare four different models: (1) model without temporal context (w/o Context); (2) model only incorporating local context (w/ Local Context); (3) model only incorporating global context (w/ Global Context); (4) model incorporating both local and global contexts (w/ Local+Global Contexts). We achieve higher mAP nearly at all IoU thresholds when incorporating either local or global context, and we can further boost the performance by combining both contexts at the same time.

4.3.4 Qualitative Results

We provide qualitative detection results on ActivityNet to demonstrate the effectiveness and robustness of our proposed DTPN. As shown in Fig. 4.4, different video streams contain very diversified background context and different activity instances vary a lot in temporal location and scale. DTPN is able to predict the accurate temporal span as well as the correct activity category, and it is also robust to detect multiple instances with various length in a single video.

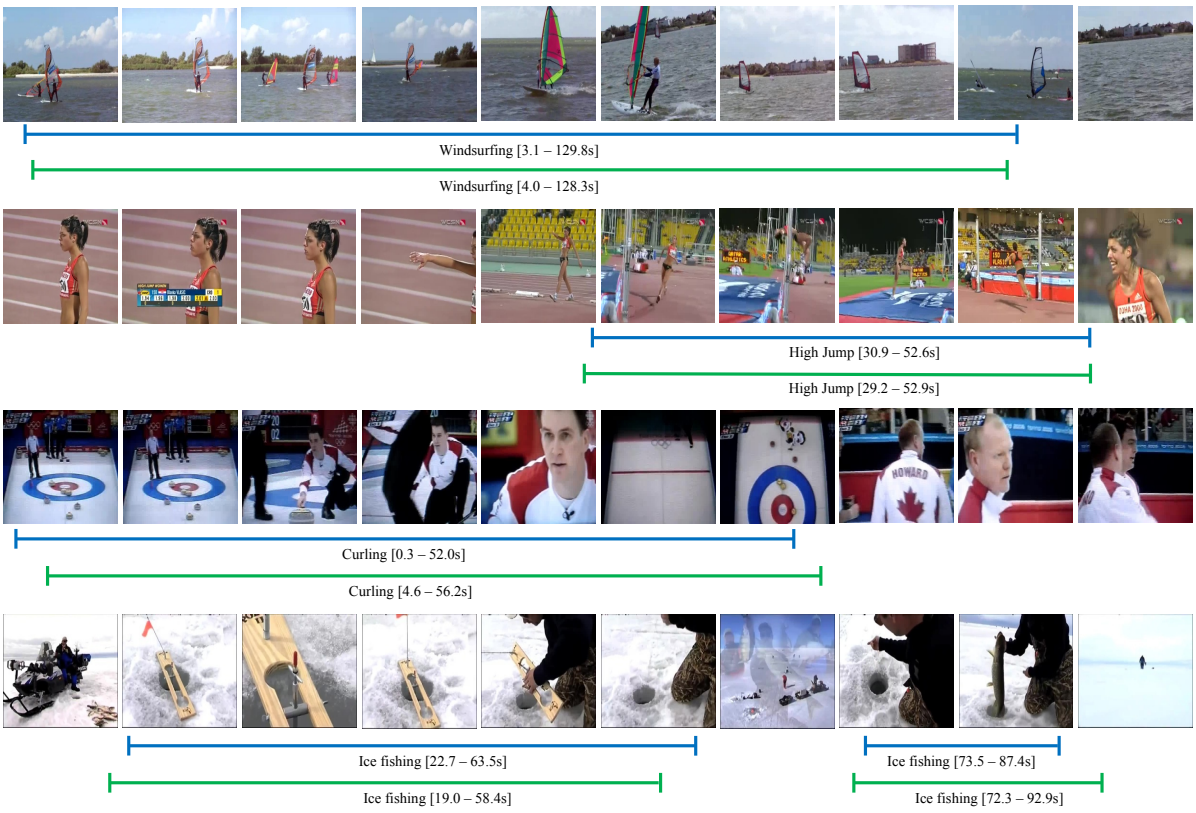

Figure 4.4: Qualitative visualization of the top detected activities on ActivityNet. Each sequence consists of the ground-truth (blue) and predicted (green) activity segments and class labels.

Activity Detection Speed. We benchmark our network on a single GTX 1080 Ti GPU to measure the activity detection speed. One activity detection in our framework is measured as a single forward-pass of the whole network, and we follow the same strategy reported in [98] to calculate the approximate detection time for different methods. In Table 4.6, we compare our approach with the state-of-the art methods in the approximate computation time to process each video. Due to the single-shot end-to-end design with simple Conv3D building blocks, our DTPN is very efficient and can process a single video in $0.5s$ which is significantly faster than most state-of-the-art methods [89, 97].

Method	Shou *et al.* [97]	Xu *et al.* [89]	Mahasseni *et al.* [98]	DTPN(ours)
Time (s)	> 930	3.2	0.35	**0.5**

Table 4.6: Comparison of our approach and the state-of-the-art methods in the approximate computation time(s) to process each video on ActivityNet dataset.

4.4 Conclusion

In this chapter, we introduce DTPN, a novel network architecture specifically designed to address three key challenges arising from the scale variation problem for temporal activity detection. DTPN employs a multi-scale pyramidal structure with three novel architectural designs: 1) pyramidal input feature extraction with dynamic sampling; (2) multi-scale feature hierarchy with two-branch network; and (3) local and global temporal contexts. We achieve state-of-the-art performance on the challenging ActivityNet dataset, while maintaining an efficient single-shot, end-to-end design.

Chapter 5

Moment Alignment Network for Natural Language Moment Retrieval

In this chapter, we strive for natural language moment retrieval in long, untrimmed video streams. The problem is not trivial especially when a video contains multiple moments of interests and the language describes complex temporal dependencies, which often happens in real scenarios. We identify two crucial challenges: semantic misalignment and structural misalignment. However, existing approaches treat different moments separately and do not explicitly model complex moment-wise temporal relations. We present Moment Alignment Network (MAN), a novel framework that unifies the candidate moment encoding and temporal structural reasoning in a single-shot feed-forward network. MAN naturally assigns candidate moment representations aligned with language semantics over different temporal locations and scales. Most importantly, we propose to explicitly model moment-wise temporal relations as a structured graph and devise an iterative graph adjustment network to jointly learn the best structure in an end-to-end manner. We evaluate the proposed approach on two challenging public benchmarks DiDeMo and Charades-STA, where our MAN significantly outperforms the state-of-the-art by a large

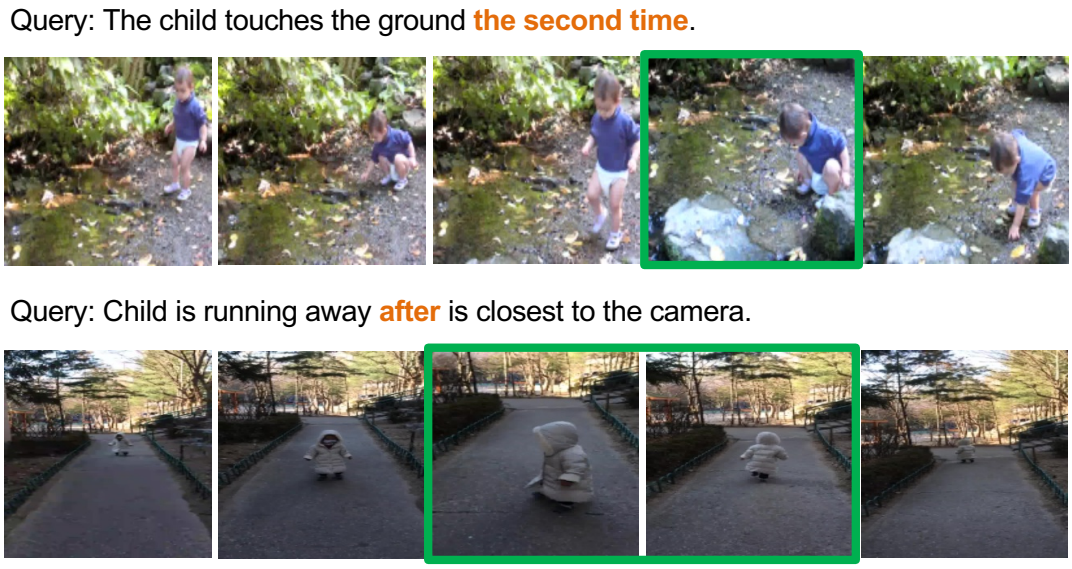

Figure 5.1: The natural language moment retrieval task in untrimmed videos. To properly localize the moment, the retrieval model must handle both *semantic misalignment (top)* with multiple moments of interests and *structural misalignment (bottom)* with complex temporal dependencies.

margin.

5.1 Introduction

Video understanding is a fundamental problem in computer vision and has drawn increasing interests over the past few years due to its vast potential applications in surveillance, robotics, etc. While fruitful progress [15, 13, 1, 16, 14, 21, 40, 99, 100, 90, 39, 97, 42, 30, 29, 89, 92] has been made on activity detection to recognize and localize temporal segments in videos, such approaches are limited to work on pre-defined lists of simple activities, such as playing basketball, drinking water, etc. This restrains us from moving towards real-world unconstrained activity detection. To solve this problem, we tackle the natural language moment retrieval task. Given a verbal description, our goal is to determine the start and end time (*i.e.* localization) of the temporal segment (*i.e.*

moment) that best corresponds to this given query. While this formulation opens up great opportunities for better video perception, it is substantially more challenging as it needs to model not only the characteristics of sentence and video but also their complex relations.

On one hand, a real-world video often contains multiple moments of interests. Consider a simple query like "The child touches the ground the second time", shown in Figure 5.1, a robust model needs to scan through the video and compare the video context to find the second occurrence of "child touches the ground". This raises the first challenge for our task: *semantic misalignment*. A simple ordinal number will result in searching from a whole video, where a naive sliding approach will fail. On the other hand, the language query usually describes complex temporal dependencies. Consider another query like "Child is running away after is closest to the camera", different from the sequence described in sentence, the "close to the camera" moment happens before "running away". This raises the second challenge for our task: *structural misalignment*. The language sequence is often misaligned with video sequence, where a naive matching without temporal reasoning will fail.

These two key challenges we identify: semantic misalignment and structural misalignment have not been solved in existing methods [43, 44] for the natural language moment retrieval task. Existing methods sample candidate moments by scanning videos with varying sliding windows, and compare the sentence with each moment individually in a multi-modal common space. Although simple and intuitive, this individualist representations of sentence and video make it hard to model semantic and structural relations among two modalities.

To address the above challenges, we propose an end-to-end Moment Alignment Network (MAN) for the natural language moment retrieval task. The proposed MAN model directly generates candidate moment representations aligned with language seman-

tics, and explicitly model temporal relationships among different moments in a graph-structured network. Specifically, we encode the entire video stream using a hierarchical convolutional network and naturally assign candidate moments over different temporal locations and scales. Language features are encoded as efficient dynamic filters and convolved with input visual representations to deal with semantic misalignment. In addition, we propose an Iterative Graph Adjustment Network (IGAN) adopted from Graph Convolution Network (GCN) [63] to model relations among candidate moments in a structured graph. Our contributions are as follows:

- We propose a novel single-shot model for the natural language moment retrieval task, where language description is naturally integrated as dynamic filters into an end-to-end trainable fully convolutional network.
- To the best of our knowledge, we are the first to exploit graph-structured moment relations for temporal reasoning in videos, and we propose the IGAN model to explicitly model temporal structures and improve moment representation.
- We conduct extensive experiments on two challenging benchmarks: Charades-STA [44] and DiDeMo [43]. We demonstrate the effectiveness of each component and the proposed MAN significantly outperforms the state-of-the-art by a large margin.

5.2 Moment Alignment Network

In this work, we address the natural language moment retrieval task. Given a video and a natural language description as a query, we aim to retrieve the best matching temporal segment (*i.e.*, moment) as specified by the query. To specifically handle the semantic and structural misalignment between video and language, we propose Moment

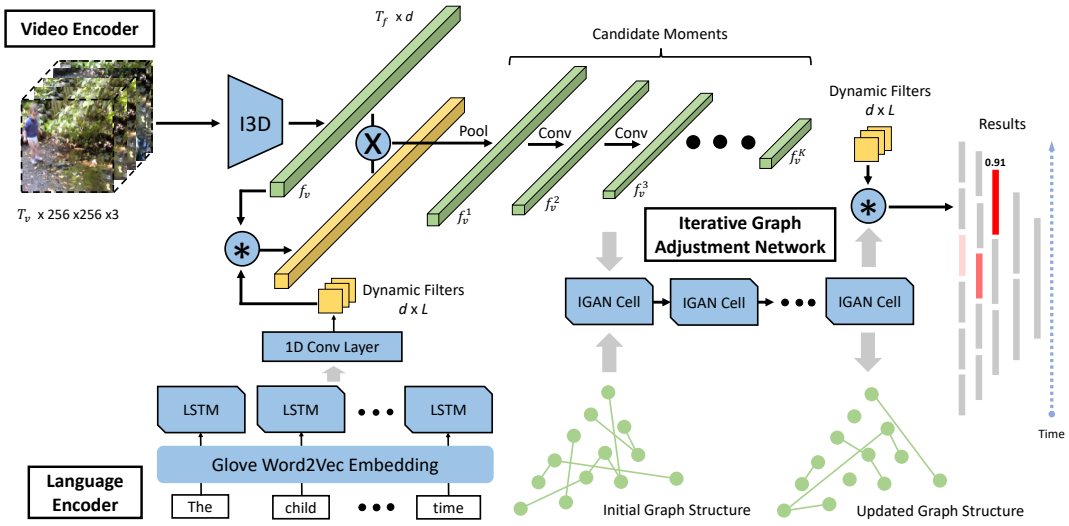

Figure 5.2: An overview of our end-to-end Moment Alignment Network (MAN) for natural language moment retrieval (best viewed in color). MAN consists three major components: (1) A language encoder to convert the input language query to dynamic convolutional filters through a single-layer LSTM. (2) A video encoder to produce multi-scale candidate moment representations in a hierarchical fully-convolutional network, where input visual features are aligned with language semantics by convolution. (3) An iterative graph adjustment network to directly model moment-wise temporal relations and update moment representations. Finally, the moments are retrieved by its matching scores with the language query.

Alignment Network (MAN), a novel framework combining both video and language information in a single-shot structure to directly output matching scores between moment and language query through temporal structure reasoning. As illustrated in Figure 5.2, our model consists of three main components: a language encoder, a video encoder and an iterative graph adjustment network. We introduce the details of each component and network training in this section.

5.2.1 Language Encoding as Dynamic Filters

Given an input of a natural language sentence as a query that describes the moment of interest, we aim to encode it so that we can effectively retrieve specific moment in video.

Instead of encoding each word with a one-hot vector or learning word embeddings from scratch, we rely on word embeddings obtained from a large collection of text documents. Specifically, we use the Glove [101] word2vec model pre-trained on Wikipedia. It enables us to model complex linguistic relations and handle words beyond the ones in the training set. To capture language structure, we use a single-layer LSTM network [102] to encode input sentences. In addition, we leverage the LSTM outputs at all time steps to seek more fine-grained interactions between language and video. We also study the effects of using word-level or sentence-level encoding in our ablation study.

In more detail, a language encoder is a function $F_l(\omega)$ that maps a sequence of words $\omega = \{w_i\}_{i=1}^{L}$ to a semantic embedding vector $f_l \in \mathbb{R}^{L \times d}$, where L is the number of words in a sentence and d is the feature dimension, and F_l is parameterized by Glove and LSTM in our case.

Moreover, to transfer textual information to the visual domain, we rely on dynamic convolutional filters as earlier used in [103, 104]. Unlike static convolutional filters that are used in conventional neural networks, dynamic filters are generated depending on the input, in our case on the encoded sentence representation. As a general convolutional layer, dynamic filters can be easily incorporated with the video encoder as an efficient building block.

Given a sentence representation $f_l \in \mathbb{R}^{L \times d}$, we generate a set of word-level dynamic filters $\{\Gamma_i\}_{i=1}^{L}$ with a single fully-connected layer:

$$\Gamma_i = tanh(W_\Gamma f_l^i + b_\Gamma) \tag{5.1}$$

where $f_l^i \in \mathbb{R}^d$ is the word-level representation at index i, and for simplicity, Γ_i is designed to have the same number of intput channels as f_l^i. Thus, by sharing the same transformation for all words, each sentence representation $f_l \in \mathbb{R}^{L \times d}$ can be converted to

a dynamic filter $\Gamma \in \mathbb{R}^{d \times L}$ through a single 1D convolutional layer.

As illustrated in Figure 5.2, we convolve the dynamic filters with the input video features to produce a semantically-aligned visual representation, and also with the final moment-level features to compute the matching scores. We detail our usage in Section 5.2.2 and Section 5.2.3, respectively.

5.2.2 Single-Shot Video Encoder

Existing solutions for natural language moment retrieval heavily relies on handcrafted heuristics [43] or temporal sliding windows [44] to generate candidate segments. However, the temporal sliding windows are typically too dense and often times designed with multiple scales, resulting in a heavy computation cost. Processing each individual moment separately also fails to efficiently leverage semantic and structural relations between video and language.

Inspired by the single-shot object detector [24] and its successful applications in temporal activity detection [92, 90], we apply a hierarchical convolutional network to directly produce multi-scale candidate moments from the input video stream. Moreover, for the natural language moment retrieval task, the visual features itself undoubtedly play the major role in generating candidate moments, while the language features also help to distinguish the desired moment from others. As such, a novel feature alignment module is especially devised to filter out unrelated visual features from language perspective at an early stage. We do so by generating convolutional dynamic filters (Section 5.2.1) from the textual representation and convolving them with the visual representations. Similar to other single shot detectors, all these components are elegantly integrated into one feed-forward CNN, aiming at naturally generating variable-length candidate moments aligned with natural language semantics.

In more detail, given an input video, we first obtain a visual representation that summarizes spatial-temporal patterns from raw input frames into high-level visual semantics. Recently, Dai *et al.* proposed to decompose 3D convolutions into aggregation blocks to better exploit the spatial-temporal nature of video. We adopt the TAN [105] model to obtain a visual representation from video. As illustrated in Figure 5.2, an input video $V = \{v_t\}_{t=1}^{T_v}$ is encoded into a clip-level feature $f_v \in \mathbb{R}^{T_f \times d}$ where T_f is the total number of clips and d is the feature dimension. For simplicity, we set f_v and f_l to have the same number of channels. While f_v should be sufficient for building advanced recognition model [89, 90, 106], the crucial alignment information between language and vision is missing specifically for natural language moment retrieval.

As such, the dynamic convolutional filters are applied to fill the gap. We convolve the dynamic filter Γ with f_v to obtain a clip-wise response map M, and M is further normalized to augment the visual feature. Formally, the augmented feature f_v' is computed as:

$$\begin{aligned} M &= \Gamma * f_v \in \mathbb{R}^{T_v \times L} \\ M_{norm} &= softmax(sum(M)) \in \mathbb{R}^{T_v} \\ f_v' &= M_{norm} \odot f_v \in \mathbb{R}^{T_v \times d} \end{aligned} \tag{5.2}$$

where $\odot$ denotes matrix-vector multiplication.

To generate variable-length candidate moments, we follow similar design of other single-shot detectors [24, 92] to build a multi-scale feature hierarchy. Specifically, a temporal pooling layer is firstly devised on top of f_v' to reduce the temporal dimension of feature map and increase temporal receptive field, producing the output feature map of size $T_v/p \times d$ where p is the pooling stride. Then, we stack K more 1D convolutional layers (with appropriate pooling) to generate a sequence of feature maps that progressively

decrease in temporal dimension which we denote as $\{f_v^k\}_{k=1}^K, f_v^k \in \mathbb{R}^{T_k \times d}$ where T_k is the temporal dimension of each layer. Thus each temporal feature cell is responsive to a particular location and length, and therefore corresponds to a specific candidate moment.

5.2.3 Iterative Graph Adjustment Network

To encode complex temporal dependencies, we propose to model moment-wise temporal relations in a graph to explicitly utilize the rich relational information among moments. Specifically, candidate moments are represented by nodes, and their relations are defined as edges. Since we gather $N = \sum_{k=1}^K T_k$ candidate moments in total each represented by a d-dimensional vector, we denote the node feature matrix as $f_m \in \mathbb{R}^{N \times d}$. To perform reasoning on the graph, we aim to apply the GCN proposed in [63]. Different from the standard convolutions which operate on a local regular grid, the graph convolutions allow us to compute the response of a node based on its neighbors defined by the graph relations. In the general form, one layer of graph convolutions is defined as:

$$H = ReLU(GXW) \tag{5.3}$$

where $G \in \mathbb{R}^{N \times N}$ is the adjacency matrix, $X \in \mathbb{R}^{N \times d}$ is the input features of all nodes, $W \in \mathbb{R}^{d \times d}$ is the weight matrix and $H \in \mathbb{R}^{N \times d}$ is the updated node representation.

However, one major limitation of the GCN applied in our scenario is that it can only reason on a fixed graph structure. To fix this issue, we introduce the Iterative Graph Adjustment Network (IGAN), a framework based on GCN but with a learnable adjacency matrix, that is able to simultaneously infer a graph by learning the weight of all edges and update each node representation accordingly. In more detail, we iteratively updates the adjacency matrix as well as node features in a recurrent manner. The IGAN model

is fully differentiable thus can be efficiently learned from data in an end-to-end manner.

In order to jointly learn the node representation and graph structure together, we propose certain major modifications to the original GCN block: (1) Inspired by the successful residual network [20], we decompose the adjacency matrix into a preserving component and a residual component. (2) The residual component is produced from the node representation similar to a decomposed correlation [107]. (3) In a recurrent manner, we iteratively accumulate residual signals to update the adjacency matrix by feeding updated node representations. The overall architecture of a single IGAN cell is illustrated in the top half of Figure 5.3 and the transition function is formally defined as:

$$\begin{aligned} R_t &= norm(X_{t-1} W_t^r X_{t-1}^T) \\ G_t &= tanh(G_{t-1} + R_t) \\ X_t &= ReLU(G_t X_0 W_t^o) \end{aligned} \tag{5.4}$$

where $X_0 = f_m$ is the input candidate moment features, R_t is the residual component derived from the output of previous cell X_{t-1}, $norm()$ denotes a signed square root followed by a L2 normalization to normalize the features, and W_t^r and W_t^o are learnable weights. Note that the candidate moment features X_0 is the output of a hierarchical convolutional network combined with language information, thus can be jointly updated with the IGAN.

In our design, the initial adjacency matrix G_0 is set as a diagonal matrix to emphasize self-relations. we stack multiple IGAN cells as shown in the bottom half of Figure 5.3 to update the candidate moment representations as well as the moment-wise graph structure. Finally, we convolve the dynamic filter Γ with the final output X_T to compute the matching scores. We further study the effects of IGAN in our ablation study.

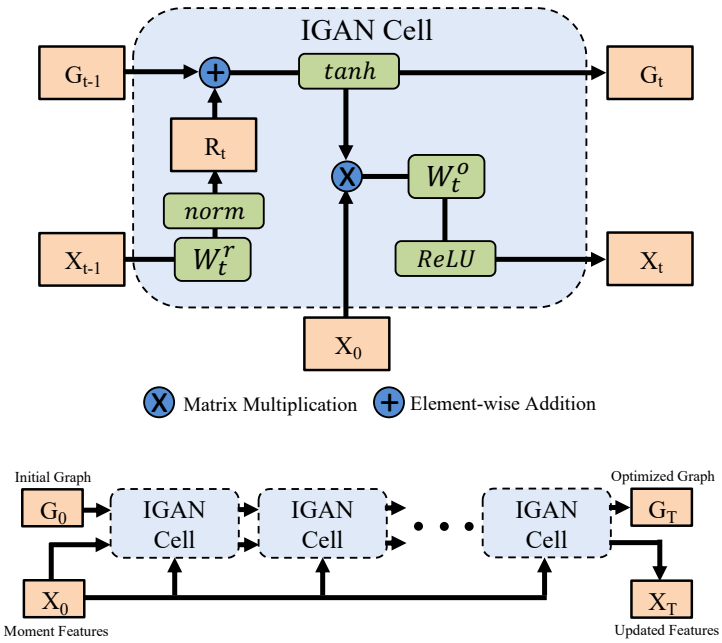

Figure 5.3: The structure of the proposed Iterative Graph Adjustment Network (IGAN). Top: In each IGAN cell, a residual component R_t is generated from the previous node representation X_{t-1} and aggregated with the preserving component G_{t-1} to produce the current adjacency matrix G_t. Node representations are updated according to Equation 5.3 with G_t, X_0 and W_t^o. Bottom: Multiple IGAN cells are connected to simultaneously update node representation and graph structure.

5.2.4 Network Training

Our training sample consists of an input video, an input language query and a ground truth best matching moment annotated with start and end time. During training, we need to determine which candidate moments correspond to a ground truth moment and train the network accordingly. Specifically, for each candidate moment, we compute the temporal IoU score with ground truth moment. If the temporal IoU is higher than 0.5, we regard the candidate moment as positive, otherwise negative. After matching each candidate moment with the ground truth, we derive a ground truth matching score s_i for each candidate moment.

For each training sample, the network is trained end-to-end with a binary classification

loss using sigmoid cross-entropy. Rather than using a hard score, we use the temporal IoU score s_i as ground truth for each candidate moment. The loss is defined as:

$$\mathcal{L} = -\frac{1}{N_b}\sum_{i}^{N_b}(s_i \log(a_i) + (1 - s_i)\log(1 - a_i)) \tag{5.5}$$

where N_b is the number of total training candidate moments in a batch, a_i is the predicted score and s_i is the ground truth score.

5.3 Experiments

We evaluate the proposed approach on two recent large-scale datasets for the natural language moment retrieval task: DiDeMo [43] and Charades-STA [44]. In this section we first introduce these datasets and our implementation details and then compare the performance of MAN with other state-of-the-art approaches. Finally, we investigate the impact of different components via a set of ablation studies and provide visualization examples.

DiDeMo The DiDeMo dataset was recently proposed in [43], specially for natural language moment retrieval in open-world videos. DiDeMo contains more than $10,000$ videos with $33,005$, $4,180$ and $4,021$ annotated moment-query pairs in the training, validation and testing datasets respectively. To annotate moment-query pairs, videos in DiDeMo are trimmed to a maximum of 30 seconds, divided into 6 segments of 5 seconds long each, and each moment contains one or more consecutive segments. Therefore, there are 21 candidate moments in each video and the task is to select the moment that best matches the query.

Following [43], we use Rank-1 accuracy (Rank@1), Rank-5 accuracy (Rank@5) and mean Intersection-over-Union (mIoU) as our evaluation metrics.

Charades-STA The Charades-STA [44] was another recently collected dataset for natural language moment retrieval in indoor videos. Charades-STA is built upon the original Charades [108] dataset. While Charades only provides video-level paragraph description, Charades-STA applies sentence decomposition and keyword matching to generate moment-query annotation: language query with start and end time. Each moment-query pair is further verified by human annotators. In total, there are $12,408$ and $3,720$ moment-query pairs in the training and testing datasets respectively. Since there is no pre-segmented moments, the task is to localize a moment with predicted start and end time that best matches the query.

We follow the evaluation setup in [44] to compute "R@n, IoU@m", defined as the percentage of language queries having at least one correct retrieval (temporal IoU with ground truth moment is larger than m) in the top-n retrieved moments. Following standard practice, we use $n \in \{1, 5\}$ and $m \in \{0.5, 0.7\}$.

Implementtation Details We train the whole MAN model in an end-to-end manner, with raw video frames and natural language query as input. For *language encoder*, each word is encoded as a 300-dimensional Glove word2vec embedding. All the word embeddings are fixed without fine-tuning and each sentence is truncated to have a maximum length of 15 words. A single-layer LSTM with $d = 512$ hidden units is applied to obtain the sentence representation. For *video encoder*, TAN [105] is used for feature extraction. The model takes as input a clip of 8 RGB frames with spatial size 256×256 and extracts a 2048-dimensional representation as output of an average pooling layer. We add another 1D convolutional layer to reduce the feature dimension to $d = 512$. Each video is decoded at 30 FPS and clips are uniformly sampled among the whole video. On Charades, we sample $T_f = 256$ clips and set the pooling stride $p = 16$ and apply a sequence of 1D convolutional filters (pooling stride 2) to produce a set of $\{16, 8, 4, 2, 1\}$ candidate moments, resulting in 31 candidate moments in total. Similarly, on DiDeMo,

Method	Rank@1	Rank@5	mIoU
TMN [61]	18.71	72.97	30.14
TGN [46]	24.28	71.43	38.62
MCN [43]	24.42	75.40	37.39
MAN(ours)	**27.02**	**81.70**	**41.16**

Table 5.1: Natural language moment retrieval results on DiDeMo dataset. MAN outperforms previous state-of-the-art methods by $\sim 3\%$ among all metrics.

Method	R@1 IoU=0.5	R@1 IoU=0.7	R@5 IoU=0.5	R@5 IoU=0.7
Random [44]	8.51	3.03	37.12	14.06
CTRL [44]	21.42	7.15	59.11	26.91
Xu *et al.* [45]	35.60	15.80	79.40	45.40
MAN(ours)	**46.53**	**22.72**	**86.23**	**53.72**

Table 5.2: Natural language moment retrieval results on Charades-STA dataset. MAN significantly outperforms previous state-of-the-art methods by a large margin.

in order to match with the pre-defined temporal boundary, we sample $T_f = 240$ clips and set pooling stride $p = 40$ with a sequence of 1D convolutional filters (pooling is adjusted accordingly) to produce a set of $\{6, 5, 4, 3, 2, 1\}$ candidate moments, resulting in 21 candidate moments in total. For both datasets, we apply 3 IGAN cells. We implement our MAN on TensorFlow [94]. The whole system is trained by Adam [109] optimizer with learning rate 0.0001.

5.3.1 Comparison with State-of-the-art

We compare our MAN with other state-of-the-art methods on DiDeMo [43] and Charades-STA [44]. Note that the video content and language queries differ a lot among two different datasets. Hence, strong adaptivity is required to perform consistently well on both datasets. Since our MAN only takes raw RGB frames as input and doesn't rely on external motion features such as optical flow, for a fair comparison, all compared

Method	Rank@1	Rank@5	mIoU
Base	23.56	77.66	36.36
Base+FA(1)	24.45	78.69	37.72
Base+FA(L)	25.10	79.57	38.78
Base+FA+IGANx1	25.67	79.36	39.13
Base+FA+IGANx2	26.10	80.08	40.21
Base+FA+IGANx3	**27.02**	**81.70**	**41.16**

Table 5.3: Ablation study for effectiveness of MAN components: Top: Advantage of a single-shot video encoder. Mid: Effectiveness of the feature alignment. Bottom: Importance of the IGAN.

methods use RGB features only.

DiDeMo Table 5.1 shows our natural language moment retrieval results on the DiDeMo dataset. We compare with state-of-the-art methods published recently including the methods that use temporal modular network [61], fine-grained frame-by-word attentions [46] and temporal contextual encoding [43]. Among all three evaluation metrics, the proposed method outperforms previous state-of-the-art methods by around 3% in absolute values.

Charades-STA We also compare our method with the recent state-of-the-art methods on Charades-STA dataset. The results are shown in Table 5.2, where CTRL [44] applies a cross-modal regression localizer to adjust temporal boundaries and Xu *et al.* [45] even boosts the performance with more closely multilevel language and vision integration. Our model tops all the methods among all evaluation metrics and significantly improves R@1, IoU=0.5 by over 10% in absolute values.

5.3.2 Ablation Studies

To understand the proposed MAN better, we evaluate our network with different variants to study their effects.

Network Components. On DiDeMo dataset, we perform ablation studies to investi-

Method	R@1 IoU=0.5	R@1 IoU=0.7	R@5 IoU=0.5	R@5 IoU=0.7
Xu *et al.* [45]	35.60	15.80	79.40	45.40
MAN-VGG	41.24	20.54	83.21	51.85
MAN-TAN	**46.53**	**22.72**	**86.23**	**53.72**

Table 5.4: Ablation study on different visual features. MAN with VGG-16 features already outperforms state-of-the-art method, and TAN features further boost the performance.

gate the effect of each individual component: single-shot video encoder, feature alignment with language query and iterative graph adjustment network.

Single-shot video encoder. In this work, we introduced a single-shot video encoder using hierarchical convolutional network for the natural language moment retrieval task. To study the effect of this architecture alone, we build a **Base** model which is the same as we described in Section 5.2.2 except for two modifications: (1) We remove the feature alignment component (Equation 5.2) and directly use the visual feature f_v to construct the network. (2) We remove all IGAN cells on top and directly feed f_m to compute matching scores. The result is reported in the top line in Table 5.3, even with only a single-shot encoding scheme, we achieve 23.56% on Rank@1 and 77.66% on Rank@5 which is better or competitive with other state-of-the-art methods.

Dynamic filter. We further validate our design to augment the input clip-level features with dynamic filters. The results are shown in the middle part in Table 5.3. On top of the Base model, we study two different variants: (1) Construct a sentence-level dynamic filter where only the last LSTM hidden state is used for feature alignment, denoted as **Base+FA(1)**. (2) Construct word-level dynamic filters where all LSTM hidden states are converted to a multi-channel filter for feature alignment, denoted as **Base+FA(L)**. We observe that Base+FA(1) already improves the accuracy compared to the base model, which indicates the importance of adding feature alignment in our model. Moreover,

adding more fine-grained word-level interactions between video and language can further improve the performance.

Iterative graph adjustment network. A major contribution of MAN is using the IGAN cell to iteratively update graph structure and learned representation. We measure the contribution of this component to the retrieval performance in the bottom section in Table 5.3, where **Base+FA+IGANxn** denotes our full model with n IGAN cells. The result shows a decrease in performance with fewer IGAN cells, dropping monotonically from 27.02% to 25.67% on Rank@1. This is because the temporal relations represented in a moment graph structure can be iteratively optimized thus more IGAN cells result in better representation for each candidate moment. Despite the performance gain, we also notice that Base+FA+IGANx3 converges faster and generalizes better with smaller variance.

Visual Features. We conduct experiments to study the effect of different visual features on Charades-STA dataset. We consider two different visual features: (1) Two-stream RGB features [15] from the original Charades dataset, which is a frame-level feature from VGG-16 [19] network, we denote the model as **MAN-VGG**. (2) TAN features as described in the paper, which is a clip-level feature from aggregation blocks, we denote the model as **MAN-TAN**. The results are summarized in Table 5.4. It can be seen that TAN features outperform VGG-16 features among all evaluation metrics, this is consistent with the fact that better base network leads to better overall performance. But more interestingly, while the overall performance using only VGG visual features is noticeably lower than using TAN features, our **MAN-VGG** model already significantly outperforms the state-of-the-art method. Since frame-level VGG-16 network provides no motion information when extracting features, this superity highlights MAN's strong ability to perform semantic alignment and temporal structure reasoning.

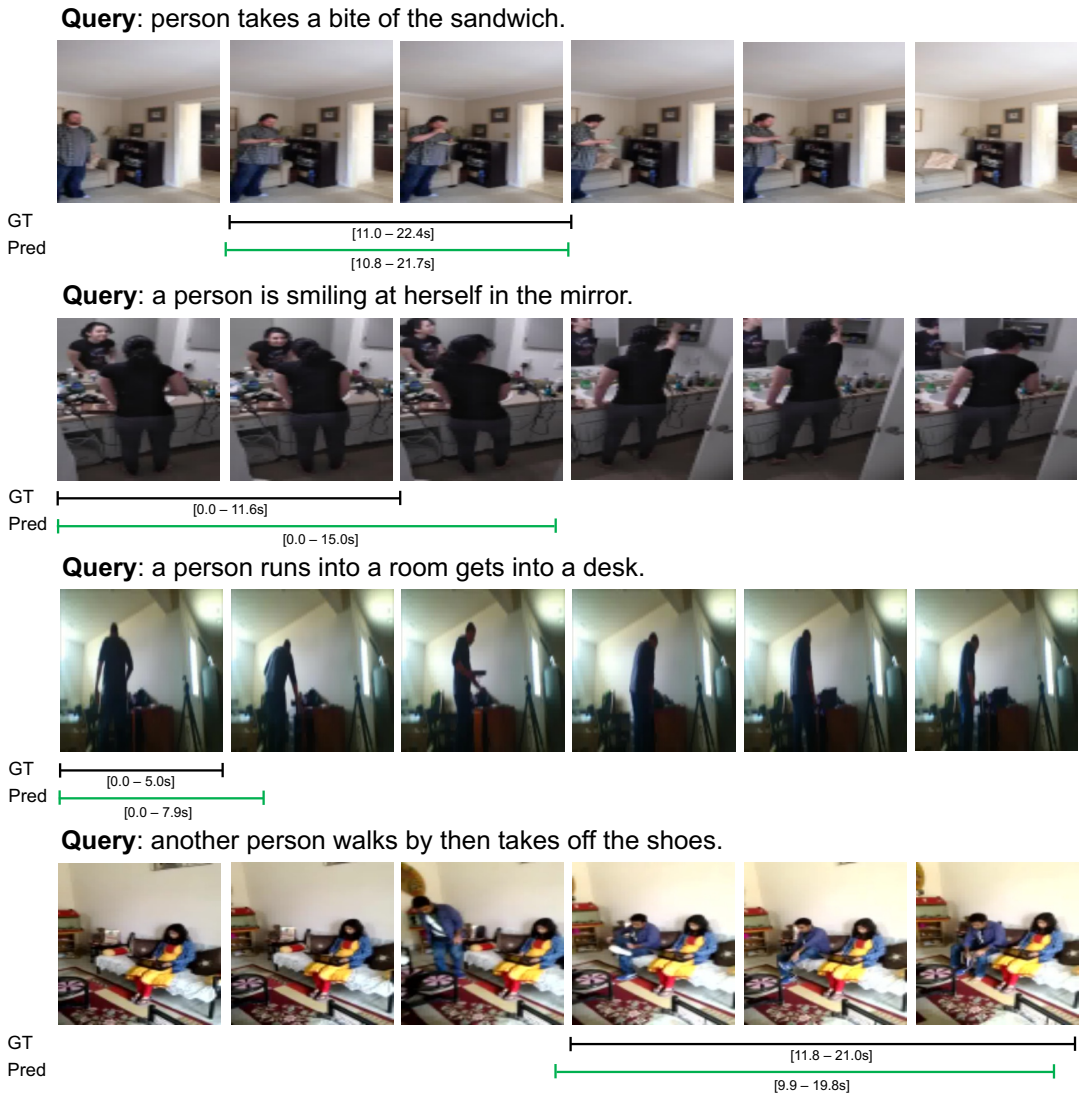

Figure 5.4: Qualitative visualization of the natural language moment retrieval results (Rank@1) by MAN (best viewed in color) on four different video-query pairs in Charades-STA dataset. Ground truth moments are marked in black and retrieved moments are marked in green. MAN is able to retrieve single moments such as "takes a bite" and "smiling" and continuous moments such as "gets into a desk" followed by "runs into a room" and "takes off the shoes" followed by "walks by".

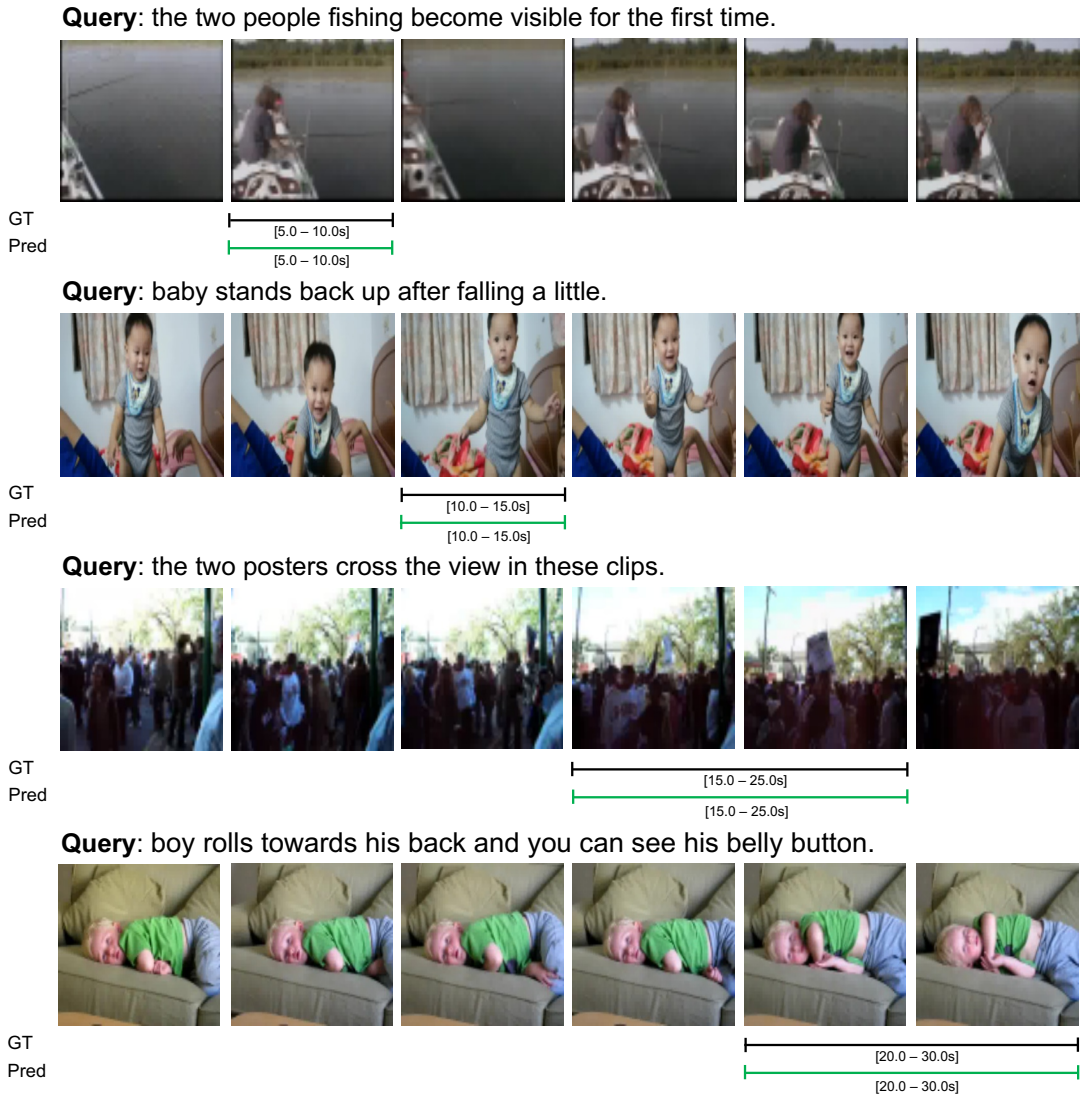

Figure 5.5: Qualitative visualization of the natural language moment retrieval results (Rank@1) by MAN (best viewed in color) on four different video-query pairs in DiDeMo dataset. Ground truth moments are marked in black and retrieved moments are marked in green. MAN is able to retrieve moments described by complex temporal dependencies such as "for the first time" and "after", and it can also distinguish the desired moments from similar unrelated contexts like correctly identify "cross the view" moment and the moment when "you can see his belly button".

5.3.3 Visualization

Qualitative Results. We provide qualitative retrieval results on Charades-STA and DiDeMo datasets. As shown in Figure 5.4 and Figure 5.5, different video streams contain very diversified contexts and different moments of interests vary a lot in temporal location and scale as well as language descriptions. On both datasets, MAN is able to retrieve the correct moment with accurate temporal boundaries. As shown in Figure 5.4, MAN can retrieve single moments such as "takes a bite" and "smiling" and it can also retrieve continuous moments such as "gets into a desk" followed by "runs into a room" and "takes off the shoes" followed by "walks by". As shown in Figure 5.5, MAN is able to retrieve moments described by complex temporal dependencies such as "for the first time" and "after", and it can also distinguish the desired moments from similar unrelated contexts like correctly identify "cross the view" moment and the moment when "you can see his belly button".

Graph Visualization. An advantage of a graph structure is its interpretability. Figure 5.6 visualizes the final moment-wise graph structure learned by our model. In more detail, Figure 5.6 displays a 30-second video where "man walks" from 10 to 30 seconds and "blocks the guitar player" from 15 to 25 seconds. MAN is able to concentrate on those moments with visual information related to "man walks across the screen". It also reasons among multiple similar moments including some incomplete moments (15-20s, 20-25s) and some other moments partially related to "blocks the guitar player" (10-20s, 10-25s) to retrieve the one best matching result (15-25s).

5.4 Conclusion

We have presented MAN, a Moment Alignment Network that unifies candidate moment encoding and temporal structural reasoning in a single-shot structure for natural

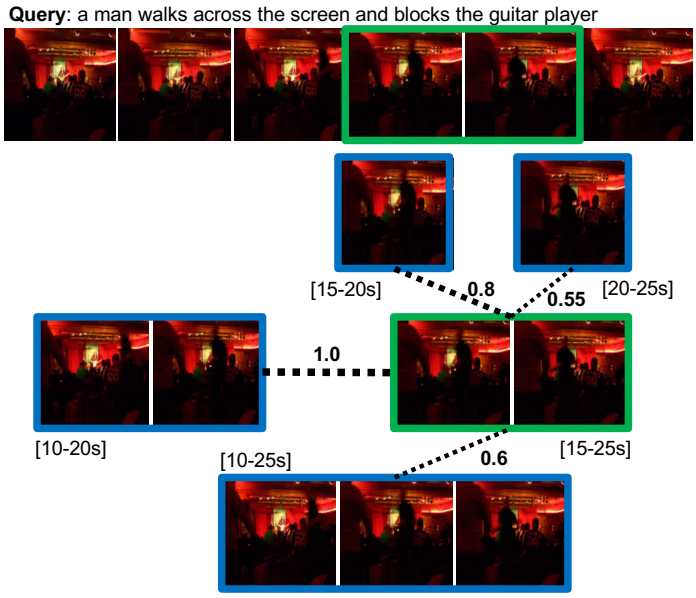

Figure 5.6: Qualitative example of MAN evaluated on a video-query pair (best viewed in color). The final moment-wise graph structure with top related edges and their corresponding moments is visualized. The retrieved moment is marked in green and other moments are marked in blue. The dashed line indicates the strength of each edge with the highest one normalized to 1.0.

language moment retrieval. Particularly, we identify two key challenges (*i.e.* semantic misalignment and structural misalignment) and study how to handle such challenges in a deep learning framework. To verify our claim, we propose a fully convolutional network to force cross-modal alignments and an iterative graph adjustment network is devised to model moment-wise temporal relations in an end-to-end manner. With this framework, We achieved state-of-the-art performance on two challenging benchmarks Charades-STA and DiDeMo.

Chapter 6

Similarity Pyramid Network for Minimum Effort Temporal Activity Localization

Existing Temporal Activity Localization (TAL) methods largely adopt strong supervision for model training which requires (1) vast amounts of untrimmed videos per each activity category and (2) accurate segment-level boundary annotations (start time and end time) for every instance. This poses a critical restriction to the current methods in practical scenarios where not only segment-level annotations are expensive to obtain but many activity categories are also rare and unobserved during training. Therefore, **Can we learn a TAL model under weak supervision that can localize unseen activity classes?** To address this scenario, we define a novel example-based TAL problem called Minimum Effort Temporal Activity Localization (METAL): Given only a few examples, the goal is to find the occurrences of semantically-related segments in an untrimmed video sequence while model training is only supervised by the video-level annotation. Towards this objective, we propose a novel Similarity Pyramid Network (SPN) that adopts the

few-shot learning technique of Relation Network and directly encodes hierarchical multi-scale correlations, which we learn by optimizing two complimentary loss functions in an end-to-end manner. We evaluate the SPN on the THUMOS'14 and ActivityNet datasets, of which we rearrange the videos to fit the METAL setup. Results show that our SPN achieves performance superior or competitive to state-of-the-art approaches with stronger supervision.

6.1 Introduction

While impressive progress has been made [15, 12, 1, 16, 14, 17, 21, 40, 35, 90, 39, 97, 30, 29, 89, 92, 110] to recognize and localize temporal segments in videos, success of these deep learning models heavily relies on the availability of a huge amount of labeled training data, meaning that model training requires the full annotation of the ground truth segment-level boundary for each activity instance among all possible classes. This severely limits their (1) scalability in practical scenarios as annotating temporal boundaries for long untrimmed videos is very expensive and time-consuming [91] and (2) applicability to newly emerging or rare events which are not observed in the original training dataset.

By contrast, human beings are capable of recognizing and localizing new activity classes in untrimmed videos by observing a few examples from each class. This motivates us to develop TAL methods that require significantly fewer annotations for training and generalize well to rare and novel activity categories. Namely, we answer the question if we can learn a TAL model under weak supervision that is able to localize unseen activity classes. Here, we introduce a new challenging example-based TAL problem called **Minimum Effort Temporal Activity Localization (METAL)**. As illustrated in Figure 6.1, we focus on the following scenario: during training, we have (1) untrimmed

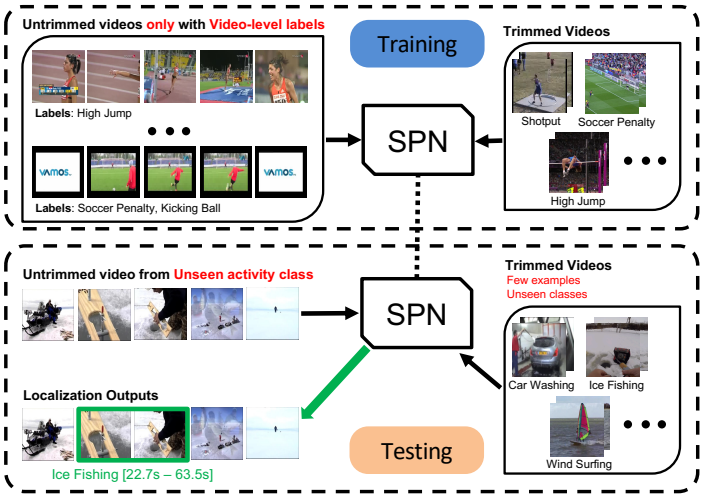

Figure 6.1: Minimum Effort Temporal Activity Localization (METAL): during training, we simply have untrimmed videos with only video-level labels and trimmed videos of the same label; during testing, the learned model is applied to TAL in untrimmed videos given only a few trimmed examples from unseen classes.

videos with only video-level labels (e.g. video tags) and (2) trimmed examples of the same labels, which are much easier to collect compared to segment-level boundary annotations. During testing, given only a few trimmed examples from unseen activity classes, we aim to localize all occurrences of semantically-related segments in the untrimmed testing videos.

The setup of METAL would greatly reduce the human efforts in developing efficient and scalable TAL methods and better simulate real-world scenario. Meanwhile, this METAL setting is also a challenging research task. First, the spatiotemporal patterns of a semantic concept in videos can have very large variations due to different environments, subjects, viewpoints, etc. Second, the model needs to localize temporal segments in untrimmed videos by comparing with trimmed examples while segment-level correspondence is not provided.

To address the above challenges, we adopt the few-shot learning technique of Relation Network [79] and propose a novel meta-learning based framework, called **Similarity**

Pyramid Network (SPN). The main idea of SPN is a *hierarchical multi-scale feature representation (similarity pyramid)* that directly measures partial similarities between an untrimmed video and trimmed examples at different temporal resolutions. To train the SPN with only video-level labels, we devise two complimentary loss functions: (1) Pairwise Content Similarity Loss (PCSL)[1] for *classification* where we compute a video-level distance metric for each pair and enforce higher similarities for positive pairs; and (2) Co-pair Structure Similarity Loss (CSSL) for *localization*, which is based on the intuition that two positive pairs should have similar distribution of similarity scores, namely higher correlation between two similarity pyramids. Thereafter, we jointly minimize the two loss functions to train the network in an end-to-end manner. The learned model is directly applied to testing videos, where the similarity pyramids are fused to yield the localization results.

Our contributions are summarized as follows:

- We introduce the METAL problem that addresses the novel task of localizing unseen activity instances in untrimmed videos given a few trimmed examples while training is only supervised by video-level labels.
- We propose a meta-learning based approach named SPN to tackle the METAL problem, which is able to measure hierarchical multi-scale similarity metrics between video pairs and simultaneously enforce classification and localization information.
- We conduct extensive experiments on two challenging benchmarks: THUMOS'14 and ActivityNet of which we rearrange the videos to fit under the METAL setup. Experimental results show that our SPN achieves performance superior or compet-

[1]In this chapter, a positive pair is defined as an untrimmed video and a trimmed video sharing the same label, while a negative pair is defined to have different labels.

itive to state-of-the-art approaches with stronger supervision.

6.2 Similarity Pyramid Network

We consider the METAL problem: Given only a few examples from unseen activity classes, the goal is to find the occurrences of semantically-related segments in an untrimmed video sequence while model training is only supervised by the video-level annotation. The setting is worth exploring as it aligns well with the practical situation: one may expect to train a localization model on dataset of easily collecting video-level labels and deploy the model to localize new activities with a few trimmed examples.

Following the few-shot learning terminologies [79, 81], we formally define the problem setup. We have three datasets: a training set, a support set and a testing set where the training set contains both untrimmed and trimmed videos with video-level labels, the support set contains labelled trimmed videos and the testing set contains untrimmed videos. The support set and testing set share the same label space, but the training set has its own label space that is disjoint with the support and testing sets. If the support set contains K trimmed examples for each of C unique classes, the target problem is called C-way K-shot.

We follow the meta-learning framework to use the *training set* during training phase and the *support set* and *testing set* during testing phase. More specifically, we follow [80, 79] to exploit the training set to mimic the few-shot learning setting via episode based training. In each training iteration, an episode is formed by randomly selecting C classes from the training trimmed videos with K samples from each of the C classes to act as the *sample set*, as well as one training untrimmed video to serve as the *query set*. This sample/query set split is designed to simulate the support/test set that will be encountered at test time. In our experiments (Section 6.3), we consider five-way one-

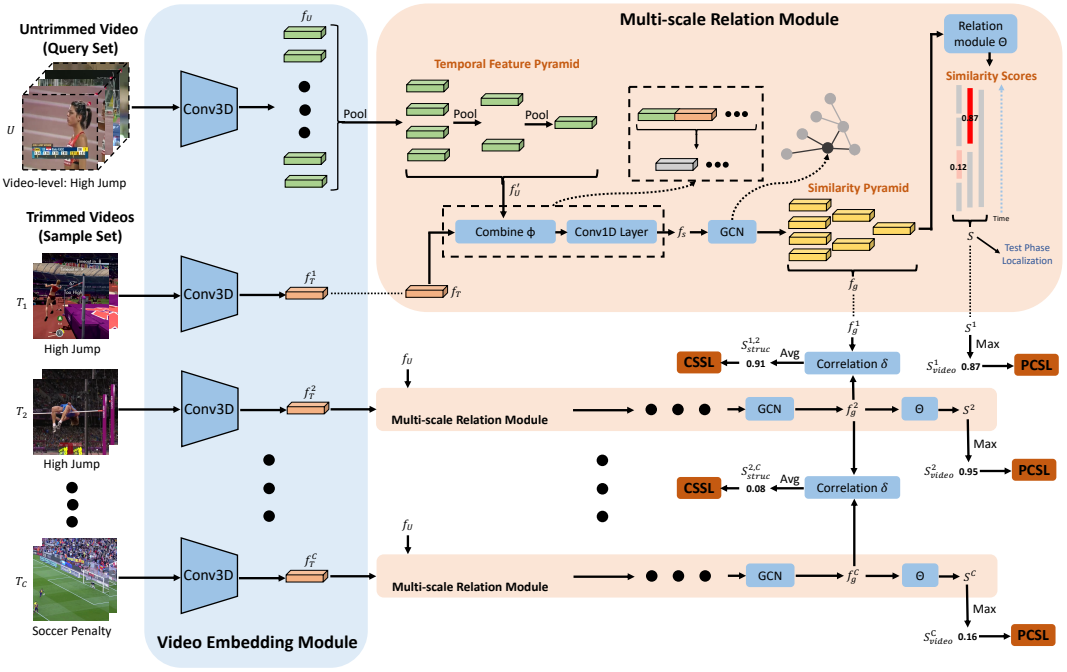

Figure 6.2: Similarity Pyramid Network (SPN) architecture for METAL under one-shot setting (best viewed in color). Both untrimmed and trimmed videos are fed into a shared Conv3D network for feature extraction, and a temporal feature pyramid is applied to summarize the untrimmed video. The features are then passed through the multi-scale relation module to obtain the similarity pyramids and similarity scores. Using these outputs, we compute two loss functions namely CSSL and PCSL, which are optimized jointly to train the network.

shot ($C = 5$, $K = 1$) and five-way five-shot ($C = 5$, $K = 5$) settings.

6.2.1 Model Overview

In this section, we present our Similarity Pyramid Network (SPN) for METAL. An overview of our proposed SPN is illustrated in Figure 6.2. First, we present the video embedding module (Section 6.2.2) that uses a shared Conv3D network to encode both untrimmed and trimmed videos, followed by a temporal feature pyramid (Section 6.2.3) to naturally summarize an untrimmed video at different temporal locations and scales. We then present the multi-scale relation module (Section 6.2.4) that directly measures the

segment-level similarities between an untrimmed video and trimmed examples. Thereafter, we introduce two loss functions PCSL and CSSL (Section 6.2.5), which we jointly optimize to learn the weights of the network. It may be noted that we compute both the loss functions using only the video-level labels. Finally, we show that the trained model can be directly applied for TAL given a few labelled examples in the support set (Section 6.2.6).

6.2.2 Video Embedding Module

In our problem setup, our SPN takes two types of input videos, namely, untrimmed video U and trimmed video T. We denote a video as a series of RGB frames $\{I_i\}_{i=1}^{F}$, where $I_i \in \mathbb{R}^{H\times W\times 3}$ is the i-th input frame and F is the total number of frames for a single video. A common practice for video processing is to use a high-quality video encoding network to extract a compact feature representation from raw frame inputs. In this work, we adopt the Res3D [14] model to obtain visual representations for both untrimmed and trimmed videos. The network weights are shared among the two different inputs.

As illustrated in Figure 6.2, the input RGB frame sequence can be represented as a tensor with dimension $\mathbb{R}^{F\times H\times W\times 3}$ where H and W are the height and width of each frame. For a trimmed video, we follow the traditional use of Res3D to uniformly sample L_T frames and obtain a fixed-dimensional 1D feature vector $f_T \in \mathbb{R}^{d_T}$ as the visual representation, where d_T is the number of output channels. For an untrimmed video, as the Res3D network can take arbitrary number of frames as input due to the fully convolutional nature, we also uniformly sample a much longer sequence of L_U frames and extract a feature map $f_U \in \mathbb{R}^{T_U\times d_U}$ as the visual representation where T_U is determined by the equivalent temporal stride of the original Res3D network. In the C-way one-shot

setting, we feed each trimmed video to the Res3D network thus generate C features for trimmed videos. For C-way K-shot where $K > 1$, we follow [79] to element-wise sum over the Res3D outputs of all samples from each class to form this class' feature representation. Thus the number of features for the sample/support set is always C in both one-shot or few-shot setting.

After the video embedding module, we extract features for both untrimmed and trimmed videos which we denote as f_U and $\{f_T^i\}_{i=1}^C$ where $f_T^i \in \mathbb{R}^{d_T}$ represents each class' feature. Note that $\{f_T^i\}_{i=1}^C$ are from C different classes during testing but not necessarily in the sample set (during training) in order to enrich the training dynamics.

6.2.3 Temporal Feature Pyramid

Although f_U serves as a good feature representation for an untrimmed video, it only summaries the video at a single temporal resolution. One may think of applying the temporal sliding window approach [81], but such method is computationally intensive and cannot model complex temporal relations. Inspired by the single-shot object detector [24] and its successful applications in temporal activity localization [92, 90], we construct a multi-scale feature pyramid to directly produce temporal features at variable scales. Unlike the previous activity localization methods trained with strong supervision, the few-shot problem setup requires us to minimize the network size to prevent overfitting. Thus, we use a simple multi-scale pooling architecture instead of multiple layers of temporal convolutions.

Specifically, we stack N_U 1D max-pooling layers with a pooling stride of 2 to generate a sequence of feature maps that progressively decrease in temporal dimension which we denote as $\{f_U^i\}_{i=1}^{N_U}$, $f_U^i \in \mathbb{R}^{T_U^i \times d_U}$ where T_U^k is the temporal dimension of each layer. Thus each temporal feature is responsive to a particular temporal location and scale. For

simplicity, we denote the final encoding feature for an untrimmed video as $f'_U \in \mathbb{R}^{N \times d_U}$ where $N = \sum_{i=1}^{N_U} T_U^i$ is the total number of temporal locations used for the multi-scale feature pyramid.

6.2.4 Multi-scale Relation Module

To learn the relations between untrimmed and trimmed videos, we follow the relation network [79] to combine the feature maps between two different inputs with operator $\Phi(f'_U, f_T)$, where f_T is a class' feature map and we omit the superscript for simplicity. Different from the relation network where only image-to-image relations are considered, we extend the formulation to video domain and deal with relations between untrimmed and trimmed videos. In this work, we assume $\Phi(\cdot, \cdot)$ to be concatenation of feature maps in depth among all temporal locations defined as:

$$f_\Phi = \Phi(f'_U, f_T) \in \mathbb{R}^{N \times d_\Phi} \tag{6.1}$$

where $d_\Phi = d_U + d_T$ is the number of channels after concatenation. We then generate a similarity embedding f_s using one single 1D convolutional (Conv1D) layer:

$$f_s = ReLU(Conv1D(f_\Phi)) \in \mathbb{R}^{N \times d_s} \tag{6.2}$$

While f_s can be directly fed into a relation module to compute the similarity scores, it only considers the content similarity at each specific temporal location. However, temporal contextual information has been proven to be critical for TAL [40, 110]. To encode such contextual relations in our network, we adopt a simple GCN on top of f_s. Different from the standard convolutions which operate on a local regular grid, the graph convolutions allow us to compute the response of a node based on its neighbors defined

by the graph connections. In this work, temporal segments are represented by nodes, and their relations are defined as edges. We use f_s as the input node features and one layer of graph convolution is defined as:

$$f_g = ReLU(Gf_sW) \tag{6.3}$$

where $G \in \mathbb{R}^{N \times N}$ is the adjacency matrix, f_s is the input feature of all nodes, $W \in \mathbb{R}^{d_s \times d_g}$ is the learnable weight matrix and $f_g \in \mathbb{R}^{N \times d_g}$ is the output node representation. In this work, we define the adjacency matrix based on the ordering of temporal segments as originally encoded in the multi-scale feature hierarchy. After one GCN layer, each node representation in f_g is enriched by the neighborhood relations. We refer to f_g as the similarity pyramid as it naturally encodes relations in a multi-scale feature pyramid.

Finally, we apply a relation module $\Theta(f_g)$ to produce similarity scores $S \in \mathbb{R}^N$ where each number is a scalar in range of 0 to 1 representing the similarity at each temporal location. In this work we assume $\Theta(\cdot)$ be a multi-Conv1D layer although other choices are possible.

6.2.5 Training

In this section, we present two proposed loss functions which use only the video-level labels as direct supervision for classification and localization, respectively. To better illustrate our idea, we consider one training batch containing one untrimmed feature f_U and C trimmed features $\{f_T^i\}_{i=1}^C$.

Pair-wise Content Similarity Loss. Here, we propose a Pair-wise Content Similarity Loss (PCSL) to add classification constraints. Considering one positive pair, although we don't know which temporal segment best corresponds to the trimmed example, it is certain that there is at least one semantically-related segment resulting in a high similar-

ity score. Similarly, all similarity scores will be small considering a negative pair. Based on this motivation, we aggregate similarity scores S to form a video-level score S_{video} via a simple max-pooling. Given a pair (f_U, f_T^i), S_{video}^i will be regressed to 1 if it is positive, otherwise 0.

Given the labels of untrimmed and trimmed videos in one batch, we formally define a positive set S_p containing all positive pairs and a negative set S_n where $|S_p| + |S_n| = C$. We define the PCSL as the sum of the sigmoid cross entropy loss for each pair:

$$\mathcal{L}_{PCSL} = -\sum_{i=1}^{C} \mathcal{L}_{sigmoid}(S_{video}^i, GT_{video}^i) \tag{6.4}$$

where S_{video}^i is the predicted video-level score, GT_{video}^i is the ground truth score $GT_{video}^i = 1, (f_U, f_T^i) \in S_p$ and $GT_{video}^i = 0, (f_U, f_T^i) \in S_n$.

Co-pair Structure Similarity Loss. While PCSL enforces the pair-wise relations between untrimmed and trimmed videos, it is location agnostic as it only measures the video-level similarity. In order to provide constraints for learning better weights for localization, we propose another Co-pair Structure Similarity Loss (CSSL). Our intuition is that given two positive pairs, for example an untrimmed video of playing basketball and two different trimmed videos of shooting, both should be matched to the same temporal region in the untrimmed sequence although the boundary annotation is unknown. To enforce such information during training, we leverage the design of similarity pyramid f_g and enforce two pyramids to have similar structures (distribution of scores) for two positive pairs.

Formally, given two positive pairs (f_U, f_T^a) and (f_U, f_T^b), we first produce the similarity pyramid after GCN as f_g^a and f_g^b respectively. Based on the above intuition, we compute the structure similarity between two similarity pyramids. Specifically, we define the

structure similarity as the average cosine similarity among all temporal locations:

$$
\begin{aligned}
S_{struc}^{a,b} &= \frac{1}{N}\sum_{i=1}^{N}\delta(f_g^a(i), f_g^b(i)) \\
\delta(f_g^a(i), f_g^b(i)) &= \frac{(f_g^a(i))^T f_g^b(i)}{||f_g^a(i)|| \cdot ||f_g^b(i)||}
\end{aligned}
\tag{6.5}
$$

where $f_g^a(i)$ and $f_g^b(i)$ indicates the feature vector at index i and $\delta(\cdot,\cdot)$ denotes the cosine similarity between two features. Note that the embeddings f_g^a and f_g^b are multi-scale similarity embeddings among different temporal locations, thus the score S_{struc} peaks when they share the same distribution. Therefore, given one positive pair, S_{struc} will be minimized when compared with another positive pair, otherwise maximized.

Given a training batch, we define the CSSL as the sum of structure similarities for every two pairs including at least one positive pair:

$$
\mathcal{L}_{CSSL} = \sum_{i=1}^{|S_p|}\sum_{j=i+1}^{|S_p|} S_{struc}^{i,j} - \sum_{i=1}^{|S_p|}\sum_{j=1}^{|S_n|} S_{struc}^{i,j} \tag{6.6}
$$

where $S_{struc}^{i,j}$ is the predicted structure similarity, $|S_p|$ is the number of positive pairs and $|S_n|$ is the number of negative pairs.

Finally, the SPN is end-to-end trainable by jointly optimizing two loss functions. The joint training allows all network weights to be trained such that the embedding module as well as the relation module are optimized for both classification and localization. The total loss is defined as:

$$
\mathcal{L} = \mathcal{L}_{PCSL} + \alpha\mathcal{L}_{CSSL} \tag{6.7}
$$

where α is used to balance the two losses.

6.2.6 Prediction

TAL via SPN is straightforward with one forward pass of the network. Considering a C-way K-shot localization problem with one untrimmed testing video and K different trimmed videos in each of the C different classes from the support set. We first extract the visual features for both untrimmed videos and trimmed videos resulting in C trimmed features and 1 untrimmed feature. Then, we compute, as the outputs of multi-scale relation module, the similarity scores S for each of the C features. For each specific temporal location, the maximum similarity score among C different classes and the corresponding class label are assigned for the temporal segment. Then the segments with low similarity score will be filtered out and the remaining segments are refined via temporal non-maximum suppression to get the final localization results.

6.3 Experiments

In this section we describe the experimental results of our method. First, we introduce the evaluation settings for the METAL setup and the implementation details of our model. Then we compare our SPN with other state-of-the-art approaches. Finally, we perform ablation studies to investigate the impact of different components of our approach and provide qualitative visualizations.

6.3.1 Dataset and Evaluation

We evaluate our SPN on two large-scale datasets, namely THUMOS'14 [11] and ActivityNet [10]. The rationale behind the choice is mainly because these are standard datasets used in activity analysis with a lot results reported on them, and hence, the choice allows us to compare our results readily with the state-of-the-art. While the original datasets

are collected for TAL with strong supervision, we rearrange the videos to fit under the METAL setup by (1) Removing the boundary annotations for untrimmed videos; (2) Splitting activity classes into mutually exclusive sets; (3) Pairing each untrimmed video with trimmed examples from different sources. We detail the evaluation settings below.

Evaluation settings. We follow the problem definition as described previously. In our experiments, we consider the five-way localization problem under one-shot ($K = 1$) and five-shot ($K = 5$) settings. During the training, in each iteration, we construct the *sample set* by randomly sampling five classes from the subset of the training classes, and then for each class we randomly sample K trimmed videos. For the *query set*, we randomly sample one untrimmed video. During the testing phase, the setup is identical to that of the training phase, only now we use the *support set* and *testing set*. Note that the support set should have at least one class overlap with the video-level label in the testing set.

We follow the conventions to report the mean Average Precision - mAP@a where a denotes the temporal Intersection over Union (tIoU) threshold, and the average mAP among 10 tIoU thresholds [0.5:0.05:0.95]. As can be easily seen, there are a large number of different combinations of the trimmed and untrimmed videos (random classes and random samples), and the performance is dependent on those choices. We follow the few-shot tradition [81, 79] to get the reliable test results, namely, we randomly sample 1000 testing batches and the final results are reported by averaging over all these batches.

ActivityNet v1.2 [10] ActivityNet is a recently released benchmark for temporal activity localization. The dataset is released in two versions, and to facilitate comparisons with previous works, we use the version 1.2 which contains 4819 and 2383 untrimmed videos in the original training and validation subsets respectively. There are 100 different activity classes and we randomly split it into 80 classes (ActivityNet-train-80) for training and 20 classes (ActivityNet-test-20) for testing. We use the video segments in ActivityNet

as the trimmed samples and we make sure that trimmed videos do not come from the same untrimmed video when pairing them together.

THUMOS'14 [11] The THUMOS'14 dataset is another widely used benchmark for activity recognition and localization. There are 2765 trimmed videos from UCF101 dataset [111] and 413 untrimmed videos of 20 different activity categories. Although this is a smaller dataset, it has several videos where multiple activities occur, thus making it even more challenging. The 20 classes are a subset of the 101 classes in UCF101. Following [81], we split the 20 classes into 6 classes for training and 14 classes for testing. The two splits are denoted as Thumos-train-6 and Thumos-test-14. The trimmed videos come from mutual classes in UCF-101 which we denote as UCF-101-6 and UCF-101-14 for training and testing respectively.

Implementation Details For the video embedding module, we train a Res3D model [14] on the Kinetics dataset [16]. Note that the few-shot problem setup requires that the classes for testing must not be present during training and we notice that there are mutual classes between Kinetics and ActivityNet or THUMOS'14, thus, those classes are excluded when we train the Res3D model. As described in Section 6.2.2, we set $L_T = 24$, $L_U = 256$ and $d_T = d_U = 2048$. On THUMOS'14, as the length of untrimmed videos is much longer, we follow common practice [89] to cut it into non-overlapping 32-second segments and use the segmented inputs. Regarding the temporal feature pyramid, we use $N_U = 5$ for ActivityNet to generate a sequence of feature maps with temporal dimension $\{16, 8, 4, 2, 1\}$ and $N_U = 3$ for THUMOS'14 to produce the features maps with temporal dimension $\{16, 8, 4\}$. We set $d_s = d_g = 512$ for the multi-scale relation module, and the relation module $\Theta(\cdot)$ is two layers of Conv1D to map feature input to similarity scores with sigmoid activation. The whole SPN network is optimized with the end-to-end loss function defined in Equation 6.7. As a speed accuracy trade-off, only the last layer of the Res3D model is jointly optimized after pre-training. We implement our SPN on

Method	Supervision	Few-shot	mAP@0.5		average mAP	
			1-shot	5-shot	1-shot	5-shot
CDC [97]	Full	Yes	8.2	8.6	2.4	2.5
Yang et al. [81]	Full	Yes	22.3	23.1	9.8	10.0
SPN (ours)	**Weak**	**Yes**	**41.9**	**45.0**	**26.5**	**28.8**
AutoLoc [72]	Weak	No	45.2		30.8	

Table 6.1: TAL results on ActivityNet v1.2 (in percentage). mAP at tIoU threshold 0.5 and average mAP are reported. Methods are categorized into three groups: Weak supervision provides video-level labels during training; Full supervision provides temporal boundary annotations during training; Few-shot refers to only a few labeled examples are available.

TensorFlow [94]. The whole network is trained by Adam [112] optimizer with learning rate 10^{-5}.

6.3.2 Comparison with State-of-the-art

As there are no existing methods for TAL under the METAL setup. we make comparisons with state-of-the-art localization models trained with *stronger* supervision. Specifically, we compare with the methods which are trained with video-level labels but not under few-shot settings [72][2], and the methods proposed for few-shot activity localization but trained with temporal boundary annotations [97, 81][3]. It should be emphasized again that results of our methods are reported under the true METAL setting which is most challenging of all.

ActivityNet v1.2 Table 6.1 shows the localization results on the ActivityNet v1.2 dataset. All the methods are categorized into three different groups based on the level of supervision. Our SPN under the one-shot setting, significantly outperforms previous fully supervised methods among all evaluation metrics, demonstrating the superior ability of

[2]Results are reported using the few-shot evaluation settings.
[3]For CDC, we use the values reported in [81]

Method	**mAP@0.5**	
	1-shot	5-shot
CDC [97]	6.4	6.5
Yang et al. [81]	13.6	14.0
SPN (ours)	**14.3**	**16.2**
AutoLoc [72]	24.5	

Table 6.2: TAL results on THUMOS'14 (in percentage). mAP at tIoU threshold 0.5 is reported. The methods are categorized into the same groups as used in Table 6.1.

our model to effectively learn good similarity metrics between different video pairs even without having access to boundary annotations. Compared to the weakly supervised method trained with more data, although our method lacks in performance for one-shot localization, we achieves competitive accuracy when more labelled data are available (*i.e.* five-shot localization). It should be noted that we still use fewer annotations compared to that of those used in [72]. Another finding is that, for the previous methods, the result differences between one-shot and five-shot settings are very small. However, our method shows a significant improvement from increasing the number of trimmed examples, which comes from the data-driven benefit of end-to-end learning in our model. We detail the effects of each proposed component in our ablation studies.

THUMOS'14 We also compare our method with the state-of-the-art approaches on THUMOS'14 dataset. The results are shown in Table 6.2 where the methods are also categorized into the same groups as used in Table 6.1. Our SPN consistently achieves superior or competitive performance compared with previous methods trained with stronger supervision. Note that THUMOS'14 is a more challenging dataset than ActivityNet for the METAL problem, as the former has much longer untrimmed videos and has more activity instances per video, making it harder to efficiently model similarities under weak supervision: on average, the THUMOS'14 training set has 15 instances per video, while the ActivityNet training set has only 1.5 instances per video. Hence, strong adaptivity

Method	mAP@0.5	average mAP
Yang et al. [81]	22.3	9.8
SPN-ImageNet	35.2	20.6
SPN-Kinetics	**41.9**	**26.5**
Base	13.2	7.2
+Feature Pyramid	30.3	18.2
+GCN	34.7	22.7
+CSSL	**41.9**	**26.5**

Table 6.3: Ablation study for different SPN components on ActivityNet. Top: Weight initialization for the embedding module. Bottom: Effectiveness of temporal feature pyramid, GCN and CSSL. Results are reported under five-way one-shot localization.

is required to perform consistently well on both datasets.

6.3.3 Ablation Studies

Weight Initialization. We conduct experiments to study the effect of different weight initialization for the embedding module. We consider two different initialization: (1) Res3D initialized from ImageNet [113] weights (simply duplicate 2D kernels to 3D) without pre-training on any video datasets, we denote as SPN-ImageNet. (2) Res3D pre-trained from Kinetics, we denote as SPN-Kinetics. The results are summarized in the top half of Table 6.3. It may be noted that our SPN-ImageNet already significantly outperforms the state-of-the-art method, highlighting SPN's strong ability to learn the temporal relations.

Network Components. On ActivityNet v1.2 dataset, we perform ablation studies to investigate the effect of each network component: temporal feature pyramid and GCN. All the experiments are conducted for five-way one-shot localization.

First, we implement a baseline model: we use the same Res3D network to extract features for both the untrimmed video and trimmed videos, instead of using a multi-

scale architecture to encode the untrimmed video, we directly apply a relation module to compute 32 relation scores which is then max-pooled and trained with video-level labels (PCSL only). As each score only represents a small duration of the entire video, we apply multi-scale sliding windows and use the maximum score for each windowed segment. The result is reported in the first row in bottom half of Table 6.3.

On top of this base model, we first add the temporal feature pyramid and leave other parts unchanged to study the effect of this component alone. The result is shown in the second row in bottom half of Table 6.3. We observed a significant performance jump improving mAP@0.5 from 13.2 to 30.3, this clearly demonstrates the advantage of using a multi-scale feature pyramid to directly summarize video content at different temporal locations and scales.

We further validate our design to use a GCN for modeling contextual relations in the multi-scale relation module. Specifically, based on the previous model, we add a GCN on top. As reported in the third row in bottom half of Table 6.3, we achieve higher mAP indicating the importance to enrich similarity by contextual relations.

CSSL. One major contribution of SPN is to add a CSSL during training to enforce localization supervision even without boundary annotations. As shown in the Table 6.3, adding the CSSL improves the mAP@0.5 from 34.7 to 41.9 and average mAP from 22.7 to 26.5. This significant improvement indicates the importance of training SPN with CSSL and supports our motivation to enforce structure similarity between two video pairs.

6.3.4 Visualization

Qualitative Results. The one-shot temporal activity localization results on THUMOS'14 and ActivityNet v1.2 are shown in Figure 6.3 and Figure 6.4, respectively. It should be noted that different video streams contain very diversified background con-

Figure 6.3: Qualitative visualization of one-shot temporal activity localization results by SPN (best viewed in color) on five different activity categories in THUMOS'14 dataset (from top to bottom): *Baseball Pitch*, *Golf Swing*, *Soccer Penalty*, *Long Jump* and *Cricket Bowling*. Ground truth activity segments are marked in blue and predicted activity segments are marked in green.

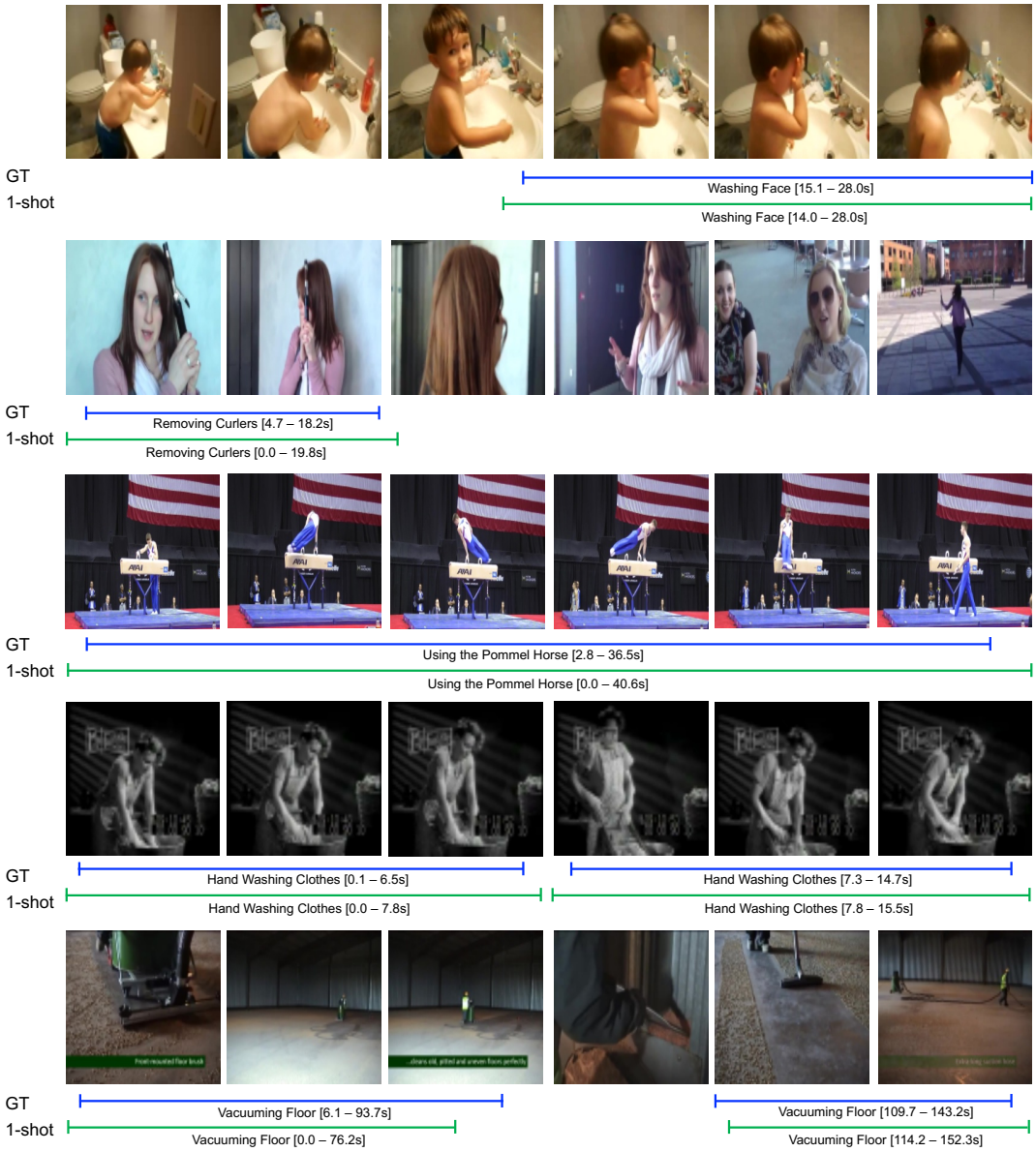

Figure 6.4: Qualitative visualization of one-shot temporal activity localization results by SPN (best viewed in color) on five different activity categories in ActivityNet v1.2 dataset (from top to bottom): *Washing Face*, *Removing Curlers*, *Using the Pommel Horse*, *Hand Washing Clothes* and *Vacuuming Floor*. Ground truth activity segments are marked in blue and predicted activity segments are marked in green.

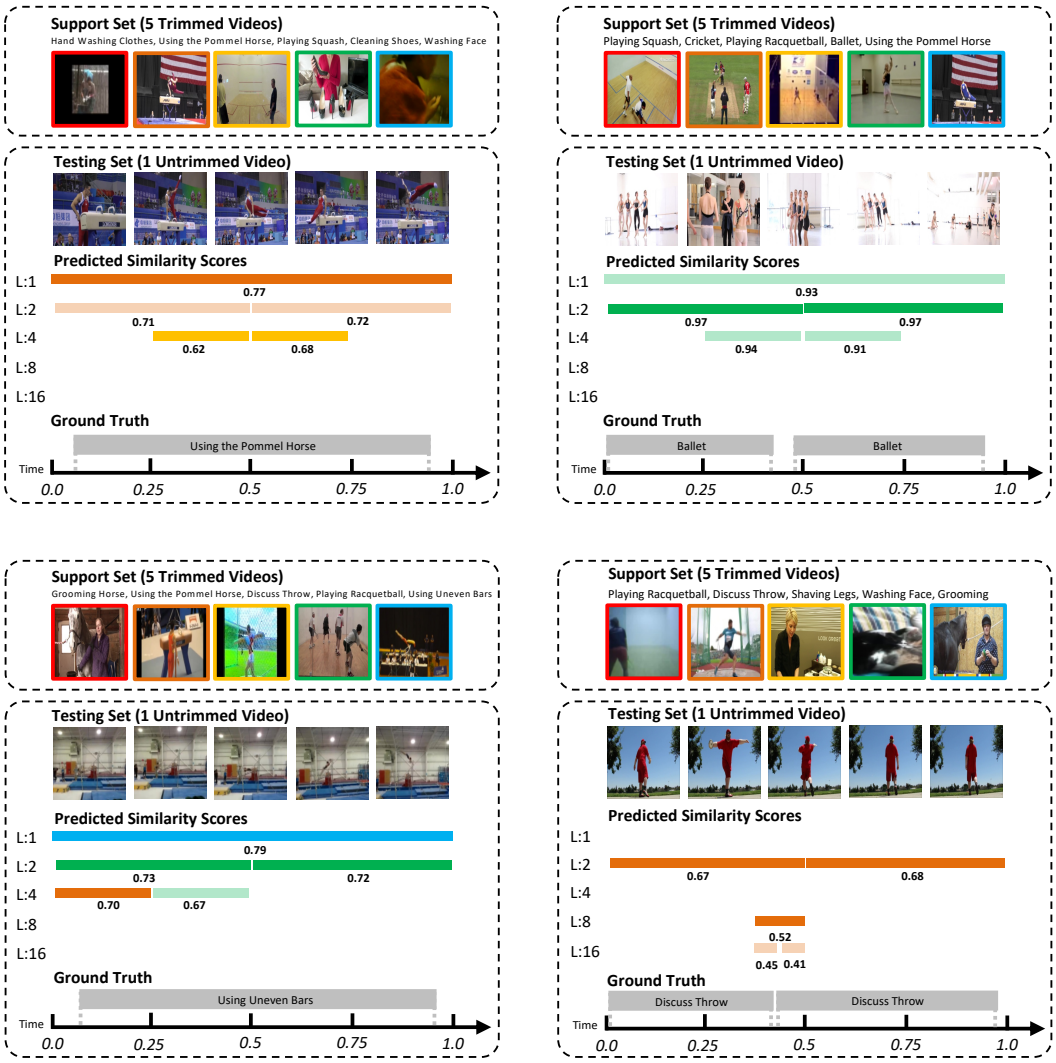

Figure 6.5: Qualitative Visualization of the multi-scale similarity scores on four different testing batches in ActivityNet v1.2 dataset (best viewed in color) under five-way one-shot localization. The segments with top 5 similarity scores are visualized with each class in the support set shown in different colors: *red*, *orange*, *yellow*, *green* and *blue*. The predicted segments are organized into a multi-scale architecture with different temporal resolutions at each layer, and the similarity score is shown under each predicted segment. Light color indicates that the corresponding segment is suppressed by temporal NMS. For better visualization, the temporal length of each video is normalized to 1.0 and a reference time line is shown at the bottom of each example.

text and different activity instances vary a lot in their temporal location and scale. In the one-shot localization setup and trained without temporal boundary annotations, our SPN is able to predict the correct activity category as well as retrieve accurate temporal boundaries. Furthermore, it is very robust to detect multiple activity instances with varying lengths in a single video.

Similarity Score Visualization. Figure 6.5 visualizes the predicted multi-scale similarity scores in ActivityNet v1.2 dataset. Under the five-way one-shot localization setting, we can see that SPN can output high similarity scores for the correct category and low similarity scores for unrelated categories. In most cases, SPN predicts multiple temporal segments for the correct category at multiple scales and only keep the best matched ones through temporal NMS. In other cases where distracted segments are observed, SPN also predicts higher similarity scores for the correct category. Although the correct trimmed video and untrimmed testing video share the same label, they differ a lot in terms of motion and appearance. Our SPN is able to produce the correct predictions most of the time, indicating the the ability of our model to learn a deep discriminative distance metric between untrimmed and trimmed videos.

6.4 Conclusion

In this chapter, we introduce a new challenging setting for TAL in untrimmed videos called Minimum Effort Temporal Activity Localization (METAL) which can also be framed as a joint problem of weakly supervised and few-shot TAL. We have presented SPN, a Similarity Pyramid Network that adapts a meta-learning framework to address the challenges in a single shot end-to-end architecture. Given only video-level labels, our SPN is end-to-end trainable by optimizing two complimentary loss functions and generalizes well to localize unseen activity classes. With this framework, although trained

under the METAL setup on the challenging THUMOS'14 and ActivityNet benchmarks, our SPN achieves performance superior or competitive to that of those state-of-the-art approaches with stronger supervision.

Chapter 7

Conclusion

In this dissertation, we have detailed our efforts in building an accurate, efficient and intelligent system for activity detection in untrimmed videos. We have demonstrated the efficacy and efficiency of a multi-scale pyramidal architecture in traditional activity detection, as well as its application to novel detection tasks such as natural language moment retrieval and minimum effort temporal activity detection by incorporating with more advanced network components. We believe the multi-scale architecture is a strong baseline for various temporal modeling problems, and it is a promising direction to learn temporal detection model guided by other modalities and provided less supervision. All of them enable significantly improvement for segment-level video understanding.

In this chapter, we summarize the contribution of our work, along with discussions about important topics in the study of segment-level video understanding.

7.1 Summary of Contributions

Segment-level video understanding has drawn increasing interests in both academic and industry communities due to its vast potential applications in security surveillance,

video analytic, videography, etc. While previous works have achieved near-perfect accuracy for temporal activity recognition, those algorithms can only classify the categories of manually trimmed videos thus not able to localize temporal segments. Temporal activity detection is for localizing and recognizing activity instances from long untrimmed video streams. It is substantially more challenging, as it is expected to handle activities with variable lengths, predicting not only the activity category but also precise temporal boundary of each instances. We propose a suite of algorithms and solutions to automatically detect segments of interest in long untrimmed videos. Three tasks are addressed towards segment-level video understanding: 1) detecting activities of interest from videos without specific purposes (*i.e.* temporal activity detection); 2) detecting temporal segment that best corresponds to a language query (*i.e.* natural language moment retrieval); 3) detecting activities given less supervision (*i.e.* weakly-supervised or few-shot activity detection).

In Chapter 3, we look at the problem of temporal activity detection. By leveraging the effective and efficient 3D convolutional kernels, we introduce S^3D, a Single Shot multi-Span Detector for temporal activity detection. We design a simple network architecture by using only a fully Conv3D network on top of the raw video frames to jointly predict the temporal boundaries as well as activity categories. A key feature of S^3D is the use of multi-scale temporal span outputs attached to multiple temporal feature maps. With this framework, we achieved state-of-the-art performance on THUMOS'14 benchmark dataset, while being efficient to run much faster than real time on a single GPU.

In Chapter 4, we further investigate how to better model the multi-scale architecture for long untrimmed videos. To this end, we introduce DTPN, a novel network architecture specifically designed to address three key challenges arising from the scale variation problem for temporal activity detection. DTPN employs a multi-scale pyramidal structure with three novel architectural designs: 1) pyramidal input feature extraction with

dynamic sampling; (2) multi-scale feature hierarchy with two-branch network; and (3) local and global temporal contexts. We achieve state-of-the-art performance on the challenging ActivityNet dataset, while maintaining an efficient single-shot, end-to-end design.

In Chapter 5, we study a more complicated problem compared to traditional temporal activity detection, namely, the task of natural language moment retrieval to locate the most related segment corresponding to a language query. We present MAN, a Moment Alignment Network that unifies candidate moment encoding and temporal structural reasoning in a single-shot structure for natural language moment retrieval. Particularly, we identify two key challenges (*i.e.* semantic misalignment and structural misalignment) and study how to handle such challenges in a deep learning framework. To verify our claim, we propose a fully convolutional network to force cross-modal alignments and an iterative graph adjustment network is devised to model moment-wise temporal relations in an end-to-end manner. With this framework, We achieved state-of-the-art performance on two challenging benchmarks Charades-STA and DiDeMo.

In Chapter 6, we introduce a new challenging setting for TAL in untrimmed videos called Minimum Effort Temporal Activity Localization (METAL) which can also be framed as a joint problem of weakly supervised and few-shot TAL. We have presented SPN, a Similarity Pyramid Network that adapts a meta-learning framework to address the challenges in a single shot end-to-end architecture. Given only video-level labels, our SPN is end-to-end trainable by optimizing two complimentary loss functions and generalizes well to localize unseen activity classes. With this framework, although trained under the METAL setup on the challenging THUMOS'14 and ActivityNet benchmarks, our SPN achieves performance superior or competitive to that of those state-of-the-art approaches with stronger supervision.

7.2 Future works

The application of segment-level video understanding is vast, and still in its infancy. We believe that in the near future, there will be tremendous opportunities for video understanding, and they will significantly improve the state-of-the-art for computer vision and artificial intelligence. In this section, I would like to address several future works that are related to my thesis work.

7.2.1 Large-scale segment retrieval

Although we have significantly advanced the natural language moment retrieval model in MAN framework, the current setting is only for retrieving the corresponding segment given one input video. Following this direction, a broader application scenario is that we want to retrieve the best matched moment in a pool of candidate long videos given a language query. While we can naively apply MAN in each single video, such method is time consuming and fail to consider cross-video correspondences. One future direction is to propose a 'Early Rejection' model to quickly filter out unrelated videos by only looking at a few number of frames, and only focus on remaining candidates for temporal localization. This direction also involves collecting new datasets for this task which can be semi-automated by cross-referencing existing benchmarks.

7.2.2 Spatial temporal video understanding

While this dissertation focuses on temporal activity detection and its variations in temporal domain, another important task is to find both spatial and temporal cues. For example, the model should output not only the temporal boundary for a specific activity but also the bounding boxes of each participants, relations among different objects, etc. While spatial domain has been extensively studies for a long time, temporal domain is

still under explored. Spatial temporal video understanding requires us to find a unified deep architecture that works in both domains. More formally, given a deep model, the architecture should be invariant to the temporal lengths of videos and predict rich spatial temporal information. Exploring such direction can further help find the common connections of spatial and temporal deep learning models, thus, benefit both domains.

Bibliography

[1] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, *Learning spatiotemporal features with 3d convolutional networks*, in *Proceedings of the IEEE international conference on computer vision*, pp. 4489–4497, 2015.

[2] A. F. Bobick and J. W. Davis, *The recognition of human movement using temporal templates*, *IEEE Transactions on pattern analysis and machine intelligence* **23** (2001), no. 3 257–267.

[3] P. Scovanner, S. Ali, and M. Shah, *A 3-dimensional sift descriptor and its application to action recognition*, in *Proceedings of the 15th ACM international conference on Multimedia*, pp. 357–360, 2007.

[4] M. D. Rodriguez, J. Ahmed, and M. Shah, *Action mach a spatio-temporal maximum average correlation height filter for action recognition*, in *2008 IEEE conference on computer vision and pattern recognition*, pp. 1–8, IEEE, 2008.

[5] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, *Action recognition by dense trajectories*, in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 3169–3176, IEEE, 2011.

[6] H. Wang and C. Schmid, *Action recognition with improved trajectories*, in *Proceedings of the IEEE international conference on computer vision*, pp. 3551–3558, 2013.

[7] F. Perronnin, J. Sánchez, and T. Mensink, *Improving the fisher kernel for large-scale image classification*, *Computer Vision–ECCV 2010* (2010) 143–156.

[8] D. Oneata, J. Verbeek, and C. Schmid, *Action and event recognition with fisher vectors on a compact feature set*, in *Proceedings of the IEEE international conference on computer vision*, pp. 1817–1824, 2013.

[9] H. Jégou, M. Douze, C. Schmid, and P. Pérez, *Aggregating local descriptors into a compact image representation*, in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 3304–3311, IEEE, 2010.

[10] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles, *Activitynet: A large-scale video benchmark for human activity understanding*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 961–970, 2015.

[11] Y. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar, *Thumos challenge: Action recognition with a large number of classes*, 2014.

[12] C. Feichtenhofer, A. Pinz, and A. Zisserman, *Convolutional two-stream network fusion for video action recognition*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1933–1941, 2016.

[13] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao, *Towards good practices for very deep two-stream convnets*, *arXiv preprint arXiv:1507.02159* (2015).

[14] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, *A closer look at spatiotemporal convolutions for action recognition*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6450–6459, 2018.

[15] K. Simonyan and A. Zisserman, *Two-stream convolutional networks for action recognition in videos*, in *Advances in neural information processing systems*, pp. 568–576, 2014.

[16] J. Carreira and A. Zisserman, *Quo vadis, action recognition? a new model and the kinetics dataset*, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4724–4733, IEEE, 2017.

[17] Z. Qiu, T. Yao, and T. Mei, *Learning spatio-temporal representation with pseudo-3d residual networks*, in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5534–5542, IEEE, 2017.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks*, in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[19] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, *arXiv preprint arXiv:1409.1556* (2014).

[20] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[21] X. Wang, R. Girshick, A. Gupta, and K. He, *Non-local neural networks*, *CVPR* (2018).

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[23] S. Ren, K. He, R. Girshick, and J. Sun, *Faster r-cnn: Towards real-time object detection with region proposal networks*, in *Advances in neural information processing systems*, pp. 91–99, 2015.

[24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, *Ssd: Single shot multibox detector*, in *European conference on computer vision*, pp. 21–37, Springer, 2016.

[25] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden, *Pyramid methods in image processing*, *RCA engineer* **29** (1984), no. 6 33–41.

[26] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, *A unified multi-scale deep convolutional neural network for fast object detection*, in *European Conference on Computer Vision*, pp. 354–370, Springer, 2016.

[27] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, *Deformable convolutional networks*, in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 764–773, 2017.

[28] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, *Feature pyramid networks for object detection*, in *CVPR*, vol. 1, p. 4, 2017.

[29] Z. Shou, D. Wang, and S.-F. Chang, *Temporal action localization in untrimmed videos via multi-stage cnns*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1049–1058, 2016.

[30] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, *Temporal action detection with structured segment networks*, in *The IEEE International Conference on Computer Vision (ICCV)*, vol. 8, 2017.

[31] A. Gaidon, Z. Harchaoui, and C. Schmid, *Temporal localization of actions with actoms*, *IEEE transactions on pattern analysis and machine intelligence* **35** (2013), no. 11 2782–2795.

[32] M. Jain, J. Van Gemert, H. Jégou, P. Bouthemy, and C. G. Snoek, *Action localization with tubelets from motion*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 740–747, 2014.

[33] P. Mettes, J. C. van Gemert, S. Cappallo, T. Mensink, and C. G. Snoek, *Bag-of-fragments: Selecting and encoding video fragments for event detection and recounting*, in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pp. 427–434, ACM, 2015.

[34] K. Tang, B. Yao, L. Fei-Fei, and D. Koller, *Combining the right features for complex event recognition*, in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2696–2703, 2013.

[35] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles, *Sst: Single-stream temporal action proposals*, in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pp. 6373–6382, IEEE, 2017.

[36] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, *Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos*, in *CVPR*, 2017.

[37] J. Gao, Z. Yang, and R. Nevatia, *Cascaded boundary regression for temporal action detection*, in *Proceedings of the British Machine Vision Conference (BMVC)*, 2017.

[38] H. Xu, A. Das, and K. Saenko, *R-c3d: Region convolutional 3d network for temporal activity detection*, in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2017.

[39] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Qiu Chen, *Temporal context network for activity localization in videos*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5793–5802, 2017.

[40] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar, *Rethinking the faster r-cnn architecture for temporal action localization*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1130–1139, 2018.

[41] T. Lin, X. Zhao, and Z. Shou, *Single shot temporal action detection*, in *Proceedings of the 2017 ACM on Multimedia Conference*, pp. 988–996, ACM, 2017.

[42] S. Buch, V. Escorcia, B. Ghanem, L. Fei-Fei, and J. Niebles, *End-to-end, single-stream temporal action detection in untrimmed videos*, in *Proceedings of the British Machine Vision Conference (BMVC)*, 2017.

[43] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell, *Localizing moments in video with natural language*, in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 5803–5812, 2017.

[44] J. Gao, C. Sun, Z. Yang, and R. Nevatia, *Tall: Temporal activity localization via language query*, in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5267–5275, 2017.

[45] H. Xu, K. He, L. Sigal, S. Sclaroff, and K. Saenko, *Multilevel language and vision integration for text-to-clip retrieval*, AAAI, 2019.

[46] J. Chen, X. Chen, L. Ma, Z. Jie, and T.-S. Chua, *Temporally grounding natural sentence in video*, in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 162–171, 2018.

[47] A. Gupta, A. Kembhavi, and L. S. Davis, *Observing human-object interactions: Using spatial and functional compatibility for recognition*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31** (2009), no. 10 1775–1789.

[48] B. Yao and L. Fei-Fei, *Modeling mutual context of object and human pose in human-object interaction activities*, in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 17–24, IEEE, 2010.

[49] J. Yao, S. Fidler, and R. Urtasun, *Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation*, in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 702–709, IEEE, 2012.

[50] X. Dai, J. Y.-H. Ng, and L. S. Davis, *Fason: First and second order information fusion network for texture recognition*, in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7352–7360, 2017.

[51] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, *Relation networks for object detection*, in *Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2018.

[52] G. G. R. G. P. Dollár and K. He, *Detecting and recognizing human-object interactions*, *CVPR* (2018).

[53] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap, *A simple neural network module for relational reasoning*, in *Advances in neural information processing systems*, pp. 4967–4976, 2017.

[54] C.-Y. Ma, A. Kadav, I. Melvin, Z. Kira, G. AlRegib, and H. P. Graf, *Attend and interact: Higher-order object interactions for video understanding*, *CVPR* (2018).

[55] B. Ni, X. Yang, and S. Gao, *Progressively parsing interactional objects for fine grained action detection*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1020–1028, 2016.

[56] B. Dai, Y. Zhang, and D. Lin, *Detecting visual relationships with deep relational networks*, in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pp. 3298–3308, IEEE, 2017.

[57] R. Hu, M. Rohrbach, J. Andreas, T. Darrell, and K. Saenko, *Modeling relationships in referential expressions with compositional modular networks*, in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pp. 4418–4427, IEEE, 2017.

[58] J. Peyre, I. Laptev, C. Schmid, and J. Sivic, *Weakly-supervised learning of visual relations*, in *ICCV 2017-International Conference on Computer Vision 2017*, 2017.

[59] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei, *Image retrieval using scene graphs*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3668–3678, 2015.

[60] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, *Graph r-cnn for scene graph generation*, *ECCV* (2018).

[61] B. Liu, S. Yeung, E. Chou, D.-A. Huang, L. Fei-Fei, and J. C. Niebles, *Temporal modular networks for retrieving complex compositional activities in videos*, in *European Conference on Computer Vision*, pp. 569–586, Springer, 2018.

[62] L. A. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell, *Localizing moments in video with temporal language*, in *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[63] T. N. Kipf and M. Welling, *Semi-supervised classification with graph convolutional networks*, in *International Conference on Learning Representations (ICLR)*, 2017.

[64] K. Marino, R. Salakhutdinov, and A. Gupta, *The more you know: Using knowledge graphs for image classification*, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20–28, IEEE, 2017.

[65] S. Yan, Y. Xiong, and D. Lin, *Spatial temporal graph convolutional networks for skeleton-based action recognition*, *AAAI* (2018).

[66] H. Bilen and A. Vedaldi, *Weakly supervised deep detection networks*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2846–2854, 2016.

[67] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, and L. Van Gool, *Weakly supervised cascaded convolutional networks*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 914–922, 2017.

[68] H. O. Song, Y. J. Lee, S. Jegelka, and T. Darrell, *Weakly-supervised discovery of visual pattern configurations*, in *Advances in Neural Information Processing Systems*, pp. 1637–1645, 2014.

[69] C. Sun, S. Shetty, R. Sukthankar, and R. Nevatia, *Temporal localization of fine-grained actions in videos by domain transfer from web images*, in *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 371–380, ACM, 2015.

[70] K. K. Singh and Y. J. Lee, *Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization*, in *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3544–3553, IEEE, 2017.

[71] L. Wang, Y. Xiong, D. Lin, and L. Van Gool, *Untrimmednets for weakly supervised action recognition and detection*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4325–4334, 2017.

[72] Z. Shou, H. Gao, L. Zhang, K. Miyazawa, and S.-F. Chang, *Autoloc: weakly-supervised temporal action localization in untrimmed videos*, in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 154–171, 2018.

[73] S. Paul, S. Roy, and A. K. Roy-Chowdhury, *W-talc: Weakly-supervised temporal activity localization and classification*, in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 563–579, 2018.

[74] C. Finn, P. Abbeel, and S. Levine, *Model-agnostic meta-learning for fast adaptation of deep networks*, in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135, JMLR. org, 2017.

[75] T. Munkhdalai and H. Yu, *Meta networks*, in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2554–2563, JMLR. org, 2017.

[76] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, *Meta-learning with memory-augmented neural networks*, in *International conference on machine learning*, pp. 1842–1850, 2016.

[77] G. Koch, R. Zemel, and R. Salakhutdinov, *Siamese neural networks for one-shot image recognition*, in *ICML Deep Learning Workshop*, vol. 2, 2015.

[78] J. Snell, K. Swersky, and R. Zemel, *Prototypical networks for few-shot learning*, in *Advances in Neural Information Processing Systems*, pp. 4077–4087, 2017.

[79] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, *Learning to compare: Relation network for few-shot learning*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018.

[80] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et. al.*, *Matching networks for one shot learning*, in *Advances in neural information processing systems*, pp. 3630–3638, 2016.

[81] H. Yang, X. He, and F. Porikli, *One-shot action localization by learning sequence matching network*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1450–1459, 2018.

[82] L. Wang, Y. Qiao, and X. Tang, *Action recognition and detection by combining motion and appearance features*, *THUMOS14 Action Recognition Challenge* **1** (2014), no. 2 2.

[83] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, *End-to-end learning of action detection from frame glimpses in videos*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2678–2687, 2016.

[84] D. Oneata, J. Verbeek, and C. Schmid, *The lear submission at thumos 2014*, .

[85] L. Wang, Y. Qiao, X. Tang, and L. Van Gool, *Actionness estimation using hybrid fully convolutional networks*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2708–2717, 2016.

[86] F. Caba Heilbron, J. Carlos Niebles, and B. Ghanem, *Fast temporal activity proposals for efficient detection of human actions in untrimmed videos*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1914–1923, 2016.

[87] R. Girshick, *Fast r-cnn*, in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[88] B. Singh and L. S. Davis, *An analysis of scale invariance in object detection–snip*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3578–3587, 2018.

[89] H. Xu, A. Das, and K. Saenko, *R-c3d: Region convolutional 3d network for temporal activity detection*, in *The IEEE International Conference on Computer Vision (ICCV)*, vol. 6, p. 8, 2017.

[90] T. Lin, X. Zhao, and Z. Shou, *Single shot temporal action detection*, in *Proceedings of the 2017 ACM on Multimedia Conference*, pp. 988–996, ACM, 2017.

[91] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, *Temporal segment networks: Towards good practices for deep action recognition*, in *European Conference on Computer Vision*, pp. 20–36, Springer, 2016.

[92] D. Zhang, X. Dai, X. Wang, and Y.-F. Wang, *S3d: Single shot multi-span detector via fully 3d convolutional network*, in *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[93] F. Yu and V. Koltun, *Multi-scale context aggregation by dilated convolutions*, *arXiv preprint arXiv:1511.07122* (2015).

[94] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et. al.*, *Tensorflow: A system for large-scale machine learning.*, in *OSDI*, vol. 16, pp. 265–283, 2016.

[95] G. Singh and F. Cuzzolin, *Untrimmed video classification for activity detection: submission to activitynet challenge*, *arXiv preprint arXiv:1607.01979* (2016).

[96] R. Wang and D. Tao, *Uts at activitynet 2016*, *AcitivityNet Large Scale Activity Recognition Challenge* **2016** (2016) 8.

[97] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, *Cdc: convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos*, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1417–1426, IEEE, 2017.

[98] B. Mahasseni, X. Yang, P. Molchanov, and J. Kautz, *Budget-aware activity detection with a recurrent policy network*, in *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

[99] X. Wang, J. Wu, D. Zhang, Y. Su, and W. Y. Wang, *Learning to compose topic-aware mixture of experts for zero-shot video captioning*, *arXiv preprint arXiv:1811.02765* (2018).

[100] X. Wang, W. Chen, J. Wu, Y.-F. Wang, and W. Y. Wang, *Video captioning via hierarchical reinforcement learning*, in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[101] J. Pennington, R. Socher, and C. Manning, *Glove: Global vectors for word representation*, in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[102] S. Hochreiter and J. Schmidhuber, *Long short-term memory*, *Neural computation* **9** (1997), no. 8 1735–1780.

[103] Z. Li, R. Tao, E. Gavves, C. G. Snoek, A. W. Smeulders, *et. al.*, *Tracking by natural language specification.*, in *CVPR*, vol. 1, p. 5, 2017.

[104] K. Gavrilyuk, A. Ghodrati, Z. Li, and C. G. Snoek, *Actor and action video segmentation from a sentence*, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5958–5966, 2018.

[105] X. Dai, B. Signh, J. Y.-H. Ng, and L. S. Davis, *Tan: Temporal aggregation network for dense multi-label action recognition*, in *WACV*, 2018.

[106] D. Zhang, X. Dai, and Y.-F. Wang, *Dynamic temporal pyramid network: A closer look at multi-scale modeling for activity detection*, *ACCV* (2018).

[107] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu, *Semantic segmentation with second-order pooling*, in *European Conference on Computer Vision*, pp. 430–443, Springer, 2012.

[108] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, *Hollywood in homes: Crowdsourcing data collection for activity understanding*, in *European Conference on Computer Vision*, 2016.

[109] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, in *International Conference on Learning Representations (ICLR)*, 2015.

[110] D. Zhang, X. Dai, and Y.-F. Wang, *Dynamic temporal pyramid network: A closer look at multi-scale modeling for activity detection*, in *The Asian Conference on Computer Vision(ACCV)*, 2018.

[111] K. Soomro, A. R. Zamir, and M. Shah, *Ucf101: A dataset of 101 human actions classes from videos in the wild*, .

[112] D. Kingma and J. Ba, *Adam: A method for stochastic optimization*, *arXiv preprint arXiv:1412.6980* (2014).

[113] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, *Imagenet: A large-scale hierarchical image database*, in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.